

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

DIPLOMOVÁ PRÁCE

**Přenos a zpracování technologických dat
bezdrátovou komunikací na bázi TNC**

**Transmission and processing the technological data
by the wireless communication based on a TNC**

Liberec 2004

David Maier

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon číslo 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce.

V Liberci dne 10. května 2004

Podpis:

Resumé

Diplomová práce se zabývá možnostmi využití bezdrátové datové sítě packet radia postavené na bázi standardního modemu typu TNC pro účely přenosu technologických dat. V realizovaném projektu byl experimentálně použit nespojovaný režim přenosu dat. První a druhá část práce se věnují úvodu do problematiky bezdrátových sítí s následnou podrobnější analýzou nízkorychlostních sítí. Důraz je kladen především na koncepci sítě packet radia a vlastnosti modemu typu TNC. V třetí části se nachází ideový návrh koncepce zamýšlené experimentální sítě a rozbor prostředků pro její realizaci. Jedná se především o popis komunikačního protokolu použitého TNC a detailní návrh formátu přenášeného datového rámce. Čtvrtá část obsahuje funkční a strukturální popis vyvinutého programového vybavení. V poslední části jsou uvedeny výsledky testů zaměřených na získání informací o chování sítě. Většina testů se orientuje na analýzu chování sítě v nespojovaném režimu.

Abstract

This thesis is focused on the possibilities of using wireless packet radio data network for technological data transmission purposes, based on a standard TNC modem. In the realized project, connectionless mode of data transfer was used as an experimental method. The first and the second parts of the thesis deal with the introduction to wireless networks, followed by more detailed analysis of low-speed networks. Above all, the conception of packet radio networks and the characters of a TNC modem are emphasized there. In the third part there are an ideological conception of the projected experimental network and the analysis of the means necessary for its realization, including a description of the communication protocol of the used TNC and a detailed concept of the transmitted data frame. The fourth part covers functional and structural specification of the developed software. The last part reveals the results of tests aimed at finding information about the network's behavior. The majority of tests are focused on the analysis of network behavior in the connectionless mode.

Obsah

POUŽITÉ ZKRATKY	8
1. ÚVOD	9
2. ZÁKLADNÍ PRINCIPY BEZDRÁTOVÉ KOMUNIKACE NA BÁZI TNC	10
2.1. HISTORIE BEZDRÁTOVÉHO PŘENOSU DAT	10
2.2. ZÁKLADNÍ PRINCIP BEZDRÁTOVÉ KOMUNIKACE	12
2.3. TOPOLOGIE NÍZKO-RYCHLOSTNÍCH BEZDRÁTOVÝCH SÍTÍ	13
2.3.1. <i>Všesměrová topologie bezdrátové datové sítě</i>	13
2.3.2. <i>Směrová topologie bezdrátové datové sítě</i>	14
2.4. FUNKCE TNC V BEZDRÁTOVÉ KOMUNIKACI	16
2.4.1. <i>Komunikační protokoly používané pro řízení TNC</i>	17
2.4.2. <i>Identifikace stanice a směrování paketu v síti CB packet-radio</i>	18
2.5. ANALÝZA MOŽNOSTÍ VYUŽITÍ TNC PRO SBĚR TECHNOLOGICKÝCH DAT Z PODRUŽNÝCH STANIC	18
2.6. ANALÝZA MOŽNOSTÍ VYUŽITÍ TNC PRO ŘÍZENÍ PODRUŽNÝCH STANIC	21
3. NÁVRH ŘEŠENÍ SÍTĚ PRO BEZDRÁTOVÝ SBĚR DAT	22
3.1. STRUČNÁ CHARAKTERISTIKA ŘÍZENÉ TECHNOLOGIE	22
3.2. ZÁKLADNÍ NÁVRH KONCEPCE SÍTĚ	23
3.3. NÁVRH METOD PRO IMPLEMENTACI V PROGRAMOVÉM VYBAVENÍ	24
3.3.1. <i>Metoda obsluhy TNC</i>	24
3.3.2. <i>Nejdůležitější ovládací příkazy režimu HOSTMODE</i>	26
3.3.3. <i>Důležité konfigurační příkazy TNC</i>	27
3.3.4. <i>Metoda filtrování přijímaných paketů</i>	28
3.3.5. <i>Metody použité při ukládání dat</i>	29
3.3.6. <i>Metoda detekce havarijních stavů</i>	29
3.3.7. <i>Metoda předávání dat mezi obslužným a vizualizačním programem</i>	30
3.4. NÁVRH KOMUNIKAČNÍHO PROTOKOLU	30
3.4.1. <i>Návrh struktury datového rámce pro sběr dat</i>	30
3.4.2. <i>Návrh protokolu pro řízení podružných stanic</i>	33
4. REALIZACE PROGRAMOVÉHO VYBAVENÍ	35
4.1. ROZHRANÍ PRO SPRÁVU A KOMUNIKACI S TNC	37

4.1.1.	<i>Problémy vzniklé při implementaci</i>	42
4.2.	PROSTŘEDKY PRO ARCHIVACI NAMĚŘENÝCH DAT	43
4.3.	VIZUALIZAČNÍ PROSTŘEDÍ	44
4.3.1.	<i>Základní vizualizační komponenty.....</i>	44
4.3.2.	<i>Hlavní část prostředí pro vizualizaci stavů podružných stanic.....</i>	48
4.3.3.	<i>Jednoduchý editor pro rozmístění vizualizačních prvků po ploše.....</i>	49
4.4.	PROGRAM SIMULACE PODRUŽNÝCH STANIC	49
4.4.1.	<i>Simulace hodnot veličin podružných stanic.....</i>	49
4.4.2.	<i>Emulace více stanic na jednom TNC.....</i>	50
5.	OVĚŘENÍ FUNKČNOSTI A MOŽNOSTÍ SÍTĚ.....	51
5.1.	VLASTNOSTI TESTOVANÉ SÍTĚ	51
5.1.1.	<i>Použité vybavení a způsob testování chování systému</i>	51
5.1.2.	<i>Závislost propustnosti sítě na přírodních podmínkách.....</i>	52
5.1.3.	<i>Vliv délky paketu na spolehlivost přenosu.....</i>	54
5.2.	VÝHODY A NEVÝHODY SBĚRU TECHNOLOGICKÝCH DAT SÍTÍ NA BÁZI TNC.....	56
6.	ZÁVĚR	58
7.	SEZNAM POUŽITÉ LITERATURY	59
8.	SEZNAM PŘÍLOH	60

Použité zkratky

<ESC>	ASCII znak odpovídající stisku klávesy „Esc“
<CR>	ASCII znaky odpovídající stisku klávesy „Enter“
API	Application Programming Interface – rozhraní pro přístup programu k prostředkům operačního systému
CB	Citizen Band – Občanská radiostanice
CSMA/CD	Carrier Sense with Multiple Access and Collision Detection
DCD	Data Carry Detect – metoda detekce smysluplného signálu
FIFO	First In First Out – typ fronty
GP	Generální Povolení – veřejná licence pro radiový provoz na vybraných frekvencích
GUI	Grafický Uživatelský Interface
LSB	Less Significant Bit
MSB	Most Significant Bit
NF	nízkofrekvenční
ODBC	Open Database Connectivity – rozhraní pro jednotný přístup k datům SQL databáze
OSI	Open Systems Interconnection – obecný model struktury programu pro přístup k síti
PDA	Pocket Digital Assistant
PLC	Programmable Logical Computer
RX	Receiver - přijímač
SSID	Secondary Station IDentifier
TCP/IP	Transmission Control Protocol/Internet Protocol
THMU	Tepelné hospodářství města Ústí nad Labem
TNC	Terminal Node Controller
TRX	Transmitter & Receiver - radiostanice
TTL	Transistor - Transistor Logic – technologie integrovaných obvodů
TUV	Teplá užitková voda
TV	Topná voda
TX	Transmitter – vysílač
UDP/IP	User Datagram Protocol/Internet Protocol
USB	Universal Serial Bus – rozhraní pro připojení periférií k osobnímu počítači
USCC	Universal Serial Connection Card – Standard pro připojení sériových modemů
VF	vysokofrekvenční

1. Úvod

Moderní přístupy k řízení průmyslových technologií se opírají o možnosti komunikace jednotlivých řídicích prvků mezi sebou, a tím se snaží optimálně využívat kapacity výrobních prostředků. V oblasti lokální komunikace existuje řada již léty ověřených a ustálených technologických řešení. Komunikační prostředky určené k překlenutí větší vzdáleností stále spadají do oblasti, kde dochází k vývoji rozličných technologií.

Tlak konkurence a trhu vede k neustálému zlepšování vlastností přenosových technologií, což mívá bohužel za následek také růst ceny řešení. Tento trend paradoxně způsobil situaci, kdy soudobé přenosové technologie mohou řádově převýšit požadavky řízené technologie na využitelnou přenosovou rychlost a kapacitu datového spoje. Nasazení takové technologie představuje pro investora neefektivně vynaloženou investici.

Prvotním impulsem pro vznik předkládané diplomové práce bylo legislativní uvolnění frekvenčních pásem k libovolnému užívání starší technologie, bezdrátové sítě packet radia, vyvinuté a po dlouhou dobu využívané především radioamatéry. Tato koncepce sítě je v řadě vlastnostech velmi unikátní a odlišná od běžně používaných komunikačních řešení.

Cílem diplomové práce je analyzovat strukturu a možnosti využití sítě packet radia pro automatizovaný přenos technologických dat. Zvoleným prostředkem komunikace je modem typu TNC a občanská radiostanice. Pro ověření možnosti nasazení sítě packet radia v oblasti průmyslových technologií a následnou analýzu jejích vlastností je třeba vytvořit komplexní softwarové zázemí umožňující u fiktivní průmyslové technologie provádět následující úkoly:

- přenášet technologická data z podružných stanic do centrálního místa,
- přenášená data archivovat pro pozdější zpracování,
- vizualizovat přenášená data,
- detekovat a okamžitě hlásit havarijní stavy.

Na základě vytvořené koncepce sítě je účelné provést testy a měření potřebná k posouzení vhodnosti nasazení sítě packet radia v průmyslu, a to zejména z hlediska spolehlivosti a celkového chování.

2. Základní principy bezdrátové komunikace na bázi TNC

2.1. Historie bezdrátového přenosu dat

Bezdrátová komunikace je jedna z možností přenosu informace z místa na místo. V dnešní době, kdy se mobilní bezdrátová komunikace dostala do popředí zájmu, je tato oblast vývoje stále zahrnuta velkým množstvím investic. V úplných počátcích bezdrátových přenosů informace, kdy se jednalo o přenos telegrafního signálu, představoval důležitou vojenskou strategickou výhodu. Díky pozdějšímu prudkému rozvoji však brzy tato technologie doznala značného rozšíření i v řadách civilního obyvatelstva. V té době se jednalo o analogové přenosy, např. radiové a televizní přenosy pro masové šíření informace a také první radiotelefonní linky pro přenos telefonních hovorů mezi městskými ústřednami. Ne všechna frekvenční pásma však vyhovovala svými vlastnostmi pro komerční nasazení, a proto ta, která nebyla vhodná, byla uvolněna pro nekomerční a radioamatérský provoz. V poslední době se hlavně díky rozvoji informační společnosti a internetu přechází od analogových přenosů k přenosům digitálním, které umožňují efektivněji využít frekvenční pásmo, zvýšit množství přenášených dat a postupně tak vytvořit globální komunikační síť.

V polovině šedesátých let 20. století začalo v souvislosti s rozvojem TTL (Transistor Transistor Logic) technologie vznikat zázemí pro počátky amatérské stavby jednoduchých počítačů [2]. Vysoké znalosti a nadšení k experimentování v oblasti konstrukce slaboproudé elektroniky měli někteří radioamatéři, kteří po zkušenostech s radiovým přenosem hlasu začali experimentovat s přenosem psaného textu po radiových vlnách. První přenosy se uskutečnily s obvody¹ a protokoly původně určenými pro dálkopis, protože charakter frekvenčního omezení a zkreslení telefonních linek byl velmi podobný radiovému přenosu. Posléze byl použit protokol X.25, u jehož návrhu se již předpokládalo nasazení v bezdrátovém provozu, avšak tvůrci předpokládali existenci pouze dvou stanic na jednom kanále. S rostoucím množstvím uživatelů však docházelo ke kolizím a bylo potřeba vyřešit metodu pro sdílení frekvenčního pásma. Nový protokol dostal název AX.25 a vyznačoval robustním zabezpečením proti chybám a kolizím vzniklým při datovém přenosu, což bylo zajištěno vyvinutím CSMA/CD algoritmu a doplněním o mechanismus ověřování a korekce

¹ Jednalo se o obvody AMD7910/7911 a TCM3105 pro normu BEL202

datového rámce. Je otázkou, zda později v roce 1975 vycházely firmy DEC, Intel a XEROX při návrhu Experimental Ethernetu rovněž z protokolu AX.25, neboť můžeme zaznamenat nápadnou podobnost ethernetového rámce [2] s I-rámcem z protokolu AX.25 [11]. Funkční ekvivalenty AX.25 U-rámce, RR-rámce a dalších nalézáme u dnešních sítí v protokolech TCP/IP a UDP/IP. Hlavním důvodem k tomuto zjednodušení mohla být realizovatelnost dobovou technologií.

V poslední době spěje vývoj bezdrátových datových technologií směrem k zlepšování širokopásmových vysokorychlostních pojitek. Jednou skupinou jsou pojítka používající ethernetové² rámce. Ačkoliv byla tato pojítka původně určena pro rozvod bezdrátové sítě uvnitř budovy, často se používají jako náhrada metalického spojení počítačových sítí a pro přístup k internetu. Díky nízkým pořizovacím nákladům však nastala situace, kdy je frekvenční pásmo určené pro tyto technologie doslova zahlcené společnostmi zajišťujícími distribuci internetu drobným firmám a domácnostem. Další skupinou jsou pojítka s krátkým dosahem sloužící pro bezdrátové připojení periférií³ k osobním počítačům, mobilním telefonům a PDA.

Nízkorychlostní a úzkopásmové sítě zůstaly na okraji zájmů, protože jsou vhodné spíše pro experimentování a osobní komunikaci mezi nadšenci než jako alternativa k metalickým počítačovým sítím. Provozování radioamatérské sítě packet-radia doznalo v České republice značného rozšíření po roce 1997 díky povolení datových provozů na vyhrazených frekvencích ve sdíleném pásmu pro všechny občany. Vznikla tím civilní obdoba této sítě, která nevyžaduje žádnou formu povolení a díky zkušenostem z amatérského provozu byla vybudována výrazně lépe oproti původní radioamatérské síti. Vzhledem k tomu, že v době, kdy se tato technologie plně uvolnila k činnostem nesouvisejícím s radioamatérstvím, byla pro případné komerční použití v řadě případů ne příliš zajímavá, nebo si již některé společnosti vybudovaly sítě vlastní koncepce v „profi“, nebo jiných pro dané účely vyhrazených pásmech, což způsobilo její opomíjení či přehlížení. Příkladem alternativních řešení mohou být sítě pro sběr meteorologických telemetrických dat, snímání obrázků z meteorologických satelitů a podobné.

² Velmi rozšířená jsou bezdrátová pojítka kompatibilní s 802.11

³ Především technologie Bluetooth a IrDa

2.2. Základní princip bezdrátové komunikace

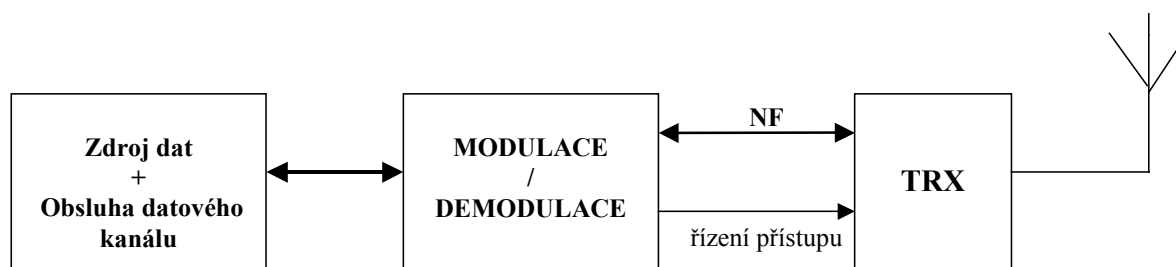
Ačkoliv se jedná o obecný model, tato kapitola se soustředí převážně na nízkorychlostní bezdrátové sítě. Bezdrátová komunikace je založena na mechanismu převodu modulovaného nízkofrekvenčního (NF) signálu do pásma s vysokou frekvencí, na kterém je možné přenášet informaci na velké vzdálenosti. Na straně příjmu se opět převádí zpět do původní formy. Blokové schéma struktury stanice pro bezdrátovou komunikaci znázorňuje obrázek 1.

Protože je frekvenční pásmo sdílené všemi stanicemi, lze komunikaci přirovnat k libovolné sběrníkové architektuře sítě. Oproti propojení dvou zařízení prostřednictvím kabelu je potřeba v případě bezdrátové komunikace řešit několik problémů vyplývajících z principu přenosu. Prvním je omezení přenosové kapacity dané maximální šířkou kanálu⁴ používaného frekvenčního pásma a omezeními vstupních filtrů TRX.

Druhým problémem je volba vhodné modulace NF signálu. Modulovaný signál je v oblasti nízkých akustických kmitočtů, které lze spolehlivě bez velkého zkreslení přenášet. Je zřejmé, že tyto dva problémy spolu úzce souvisí. Omezení vyplývají z Shanon-Kotelnikova teorému a také platí pravidlo, že se zvyšující se složitostí modulace klesá její odolnost vůči šumům. Je vhodné najít kompromis mezi složitostí modulace, přenosovou rychlostí a frekvenčním zdvihem. V praxi se osvědčily modulace AFSK pro pásma s vyšší mírou šumu, Manchester pro pásma s nižší mírou šumu a FSK, která se používá pro linky s vyšší přenosovou rychlostí, šířkou pásma s velmi nízkou hladinou šumu [1].

Vlastní převod mezi NF signálem a vysokofrekvenčním (VF) pásmem zajišťuje zařízení obecně označované jako TRX (Transmitter & Receiver). V nízkorychlostních sítích jím je nejčastěji průmyslově vyráběná radiostanice, avšak existují řešení, kdy je modul TRX integrován do jednotky modemu. K vyzáření VF signálu z koncového stupně TRX slouží anténa. Kvalitní a správně naladěná anténa představuje v celém systému prvek, který svými vlastnostmi nejvíce ovlivňuje kvalitu a sílu přenášeného signálu. Jednotlivá data se přenášejí po balíčcích, které se označují jako pakety. Z toho důvodu se často používá anglický termín packet radio, který vhodně vystihuje způsob přenosu.

⁴ Častěji se setkáme s termínem frekvenční zdvih, který je však vázán na princip modulace VF části.



Obr. 1 Blokové schéma struktury stanice pro bezdrátovou komunikaci [1]

2.3. Topologie nízko-rychlostních bezdrátových sítí

Bezdrátové sítě mají díky charakteru přenosového média oproti běžným metalickým sítím svá specifika. Jednou z nich jsou vznikající topologie sítí, které v některých ohledech nenajdou ve světě počítačových sítí obdobu. Důsledky v podobě nutné rozdílnosti algoritmů pro směrování paketu a zamezení kolizí budou lépe patrné z následujících dvou koncepcí.

2.3.1. Všesměrová topologie bezdrátové datové sítě

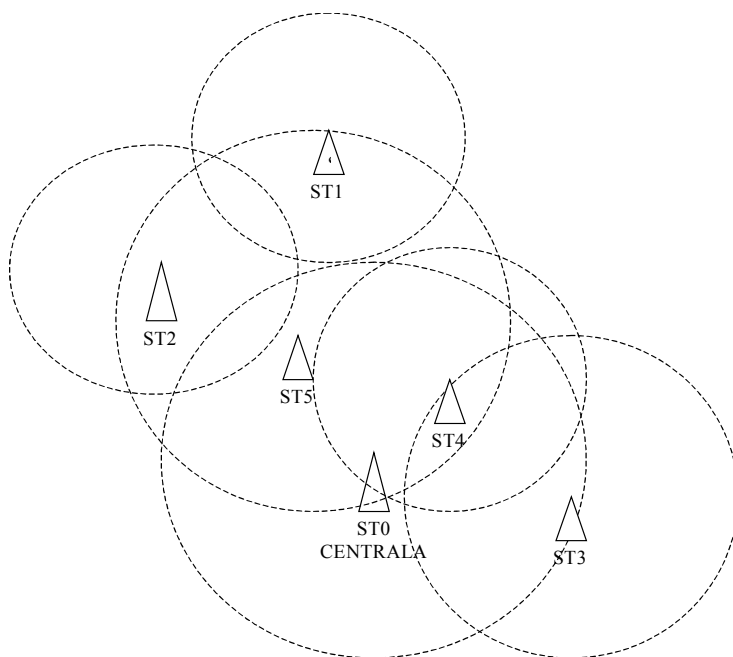
Všesměrová topologie bezdrátové datové sítě se ujala zejména v oblastech občanských a amatérských sítí. Topologie sítě packet.radio je závislá na umístění jednotlivých stanic a jejich vzájemném dosahu. Vzhledem k tomu, že stanice jsou většinou v soukromých rukách, bývá jejich činnost často omezena pouze na dobu po kterou účastník využívá některou ze služeb sítě. Struktura komunikačního protokolu proto umožňuje nastavit kompletní směrování přímo v hlavičce datového rámce.

Hranice dosahu jednotlivých stanic jsou silně závislé na kvalitě antény, pozici stanice, charakteru terénu, počasí, sluneční aktivitě a dalších vlivech. V praxi se však může pohybovat spolehlivá vzdálenost, na kterou je spojení spolehlivé nezávisle na okolním prostředí, v řádu jednotek až několika desítek kilometrů. V městském prostředí je průměrný dosah jednotlivých stanic okolo 15 km, proto je tato topologie vhodná do menších oblastí o rozměru většího města. Jednotlivé stanice jsou si navzájem rovny a komunikační protokol neupřednostňuje žádnou z nich. V případě, že je stanice mimo přímý dosah a odvysílá relaci, tak přestože jí posluchači nemusí rozumět, nedochází z jejich strany k odvysílání relace. Mechanismus snižuje pravděpodobnost vzniku kolize. V uvedeném obrázku 2 kružnice kolem vysílačů reprezentují spolehlivý dosah příjmu i vysílání. V praxi to znamená, že možná cesta od

stanice ST0 k ST1, nebo ST2 vede přes ST5, přičemž k ST4, ST3, nebo ST5 je možné se připojit přímo.

Výhody všesměrové topologie vychází ze značné flexibility sítě. V případě výpadku jedné stanice v datové cestě, je možné najít několik dalších cest k cílové stanici. Dále se jedná o levnější řešení, neboť odpadají problémy se zaměřováním a tvorbou

datových tras. Naopak nevýhodami této topologie jsou větší náročnost na kvalitu použitého komunikačního protokolu, větší redundance přenášených dat a větší náročnost na případné automatické nalezení nové cesty.



Obr. 2 Všesměrová topologie

V případě komerčního použití může nastat potřeba utajení přenášených dat a zabezpečení před ovlivňováním provozu třetí stranou. Prozatím jsem narazil pouze na jedinou realizovanou čistou formu utajení přenášených dat. Je realizována formou komprese a implementována v terminálu Xpacket. V úvahu však připadají i alternativní metody v podobě mírné změny modulace datového signálu⁵. Toto opatření má však za následek nekompatibilitu zařízení s ostatními uživateli přenosového pásma.

2.3.2. Směrová topologie bezdrátové datové sítě

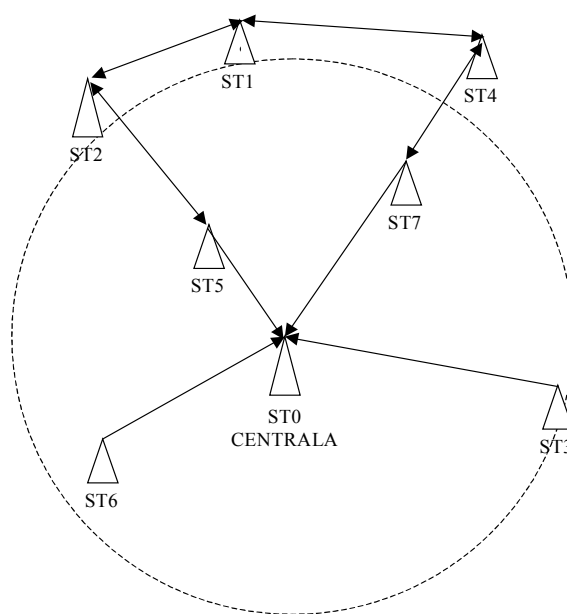
Směrová topologie bezdrátové datové sítě nalezla rozsáhlejší využití převážně v „profil“ pásnu a vyžaduje používání směrových antén. Hlavní důvody jsou nižší povolený vyzařovaný výkon stanic a přenosové vlastnosti šíření radiových vln vyhrazeného pásma. Jednou z mnoha možností, jak ji realizovat, je struktura používaná společnostmi SČE, a.s.⁶

⁵ U modulace AFSK je možné změnou krystalového oscilátoru modemového obvodu snadno „posunout“ modulační frekvence.

⁶ Severočeská Energetika a.s.

s centrálou v Děčíně, nebo společností Sazka, a.s. mající své terminály takřka v každé trafice a centrum většinou na jednom místě ve městě.

Tato topologie, znázorněná na obrázku 3, vychází z předpokladu centrálního vysílače majícího všesměrovou anténu s vyšším výkonem, který přijímá stanice, jež nejsou příliš daleko. Pokud je nějaká stanice mimo dosah centrální stanice, vytváří se uzavřené okruhy z důvodu zálohy v případě výpadku jednoho spoje. Uspořádání sítě je vhodné pro rozsáhlejší oblasti a sítě u nichž je neměnná poloha jednotlivých stanic.

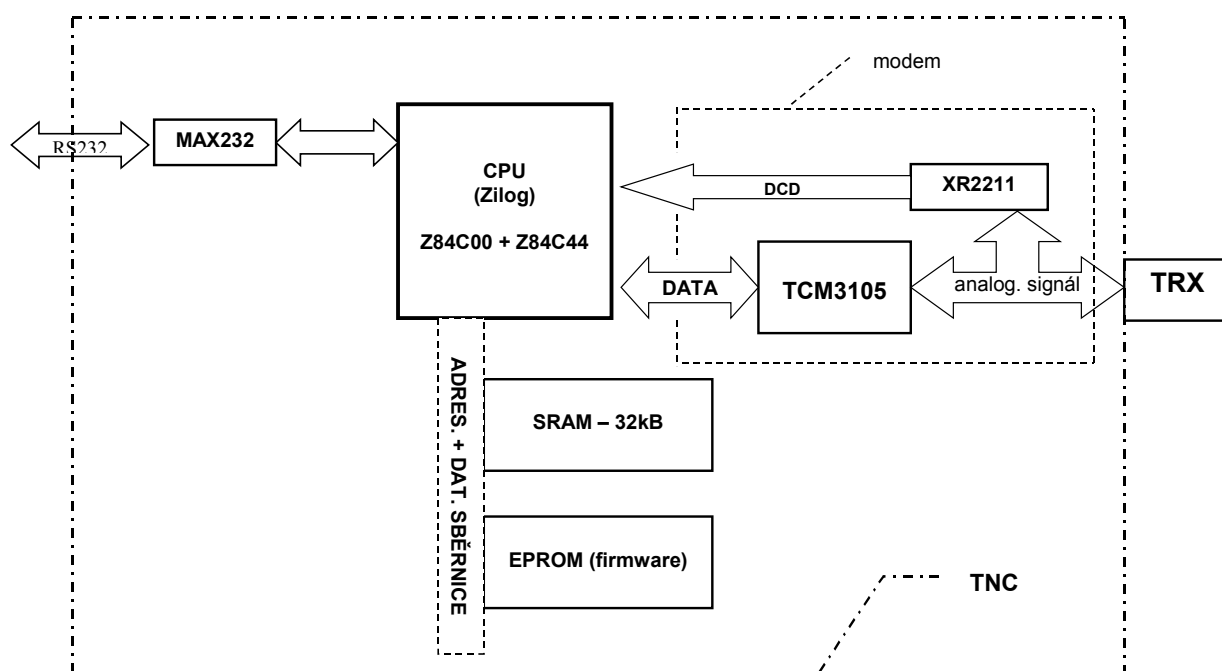


Obr. 3 Směrová topologie s centrálním uzlem

Výhodami směrové topologie jsou výrazně lepší zabezpečení proti odposlechům a přeslechům zajištěné směrovostí antén, jednoznačně definované směrování a nižší nároky na robustnost zabezpečení přenosového protokolu dané omezením vzniku možných kolizí z důvodu centrálně řízeného toku dat. Hlavní nevýhody lze spatřit v delší době nutné k vyřízení komunikace se stanicí v okruhu, dále v minimální flexibilitě sítě, což snadno způsobí odpojení její části v případě poruchy dvou spojů v jednom komunikačním okruhu. Další nevýhodou je vyšší celková cena řešení daná potřebou zdvojení komunikačních prvků a pracností spojenou s tvorbou datových tras.

2.4. Funkce TNC v bezdrátové komunikaci

TNC (Terminal Node Controller) představuje nejběžnější typ modemu [1] používaný v prostředí sítě packet-radio. Vytváří rozhraní mezi sériovou linkou RS232 a nízkofrekvenčním modulovaným signálem, který se posléze prostřednictvím TRX přenáší po rádiových vlnách. Vnitřně je tvořen dvěma logickými bloky. Modemová část převádí signál mezi NF modulovaným signálem a TTL sériovým signálem, čímž reprezentuje fyzickou vrstvu v OSI modelu. Použité TNC obsahuje modem s modulací AFSK, modulační rychlostí 1200bps a hardwarovým DCD (Data Carry Detect), což je metoda detekce smysluplného signálu.



Obr. 4 Strukturální schéma TNC-2

Nadřazený mikrokontrolér se zabývá dekódováním toku jedniček a nul vycházející z modemu. Pokud tento tok odpovídá komunikačnímu protokolu, je zpracován a výsledek dané operace se uloží do vyrovnávací paměti. Podrobnou strukturu protokolu AX.25 je možné najít ve literatuře [11]. Je důležité uvést, že tento protokol je navržen pro provoz ve spojovém režimu. Z toho vycházejí následující uživatelské vlastnosti. TNC obsahuje 1 monitorovací a 10 komunikačních kanálů. Komunikační kanály jsou virtuální. Je potřeba inicializovat žádostí o vytvoření spojení a vytvoří se až v momentě potvrzení o navázání spojení. Na těchto kanálech jsou zajištěny veškeré vlastnosti protokolu AX.25. Nejdůležitější vlastností je

spolehlivé doručení dat cílové stanici ve stejném pořadí, jako byly předány TNC, a je zajištěná různými ověřovacími rámci a případným zasláním chybějící části.

Speciálním typem kanálu je monitorovací kanál, jenž umožňuje sledovat veškerou komunikaci, probíhající na pásmu. V tomto monitorovacím kanálu je možné vyslat jakýkoliv datový rámec, který však bude vyslán v nespojovaném režimu a označen příznakem UI⁷. V praxi to znamená přímý přístup k linkové vrstvě a vypnutí veškerých mechanismů pro spolehlivé zajištění doručení datového rámce cílové stanici.

Pro uživatelskou obsluhu TNC nadřazeným systémem se používá většinou sériová linka RS232, ale objevují se i jiná řešení, jako je paralelní port, USCC karta, USB, nebo síť Ethernet.

2.4.1. Komunikační protokoly používané pro řízení TNC

Většina programů podporuje obsluhu TNC přes komunikační protokol kompatibilní s TheFirmware v režimu HOSTMODE, ale ne všechny typy TNC jej podporují. Na základě složitosti modemu se také ujalo několik jednodušších protokolů, které jsou posléze softwarově upraveny na protokol HOSTMODE. Mezi nejčastěji používané patří protokoly KISS a 6pack. Podrobný popis protokolu HOSTMODE obsahuje zdroj [14]. Ve stručnosti lze říci, že je striktně založen na principu „odpověz na dotaz“ a přestože má poměrně složitou strukturu, je vhodný pro počítačové zpracování, protože mechanismus obsluhy je algoritmizovatelný.

Původní určení TNC však předpokládalo připojení zařízení k jednoduchému sériovému terminálu, proto je v protokolu TheFirmware obsažen také jednoduchý uživatelský terminálový režim TERMINAL-MODE, plynule reagující na vstupy ze sériového terminálu. V případě že stanice obdrží nový příchozí paket, okamžitě jej zobrazí uživateli. Uživatel má možnost přepínat aktivní kanál, a několik základních příkazů pro obsluhu TNC. Ačkoliv se na první pohled zdá, že emulovat lidské chování u tohoto jednoduchého terminálu není složité, objevily se při pokusu o SW implementaci tohoto režimu problémy s rozlišováním charakteru příchozích dat vzniklé smícháním datových řetězců se systémovými hlášeními. Tato cesta by

⁷ Zkratka z anglického unnumbered info, čili nečíslovaná informace. Používá se pro broadcasty a hlášení radiomajáků [1].

vyžadovala nejspíše pokročilejší algoritmy z oblasti umělé inteligence a počítačové zpracování textových řetězců.

2.4.2. Identifikace stanice a směrování paketu v síti CB packet-radio

Protokol AX.25 pro jednoznačnou identifikaci stanice v síti používá volací znak složený z názvu a rozlišujícího čísla označovaného termínem SSID. Název se skládá z šesti znaků, přičemž doporučení k jeho sestavení se nachází v Generálním Povolení. SSID⁸ je od názvu odděleno pomlčkou, představuje ekvivalent přístupového portu. Slouží k oddělení dvou různých spojení mezi dvěma stanicemi. Struktura názvu je vytvořena v takovém tvaru, aby byla srozumitelná pro člověka, a kromě jednoznačné identifikace v rámci sítě nemá vliv na směrování paketu.

Směrování paketu je definováno při zadávání žádosti o spojení. Platí zásada, že na prvním místě je cílová stanice a za ní následuje přesný seznam stanic definující cestu, kudy má paket projít. Jednotlivé stanice se od sebe oddělují mezerou. Spojení ze stanice ST0 do ST1 dle obr.2 má směrovací řetězec „ST1 ST5 ST2“. Protokol AX.25 umožňuje zadat maximálně 8 směrovacích stanic. V případě, že je potřeba realizovat delší cestu, je zde vyřešen alternativní mechanismus. Vychází z původní topologie sítě a rozlišení stanic na tzv. nody a koncové uživatele. Nody tvoří uzly na páteřní síti a slouží jako speciální druh opakovačů, z nichž je možno vytvořit další spojení. V síti PR je v dnešní době značná část páteřní sítě přesunuta na globální síť internet.

2.5. Analýza možností využití TNC pro sběr technologických dat z podružných stanic

Vzhledem k rozsáhlým možnostem daným protokolem AX.25 je možné za specifických podmínek rozdělit celý problém na dvě různé filosofie přístupu k úloze přenášení dat – spojovaný režim a nespojovaný režim.

⁸ Rozsah SSID je 1 až 15, což teoreticky omezuje realizovat mezi dvěma stanicemi až 16 paralelních spojení. Vzhledem k možnostem modemů však většinou není možné tolik spojení realizovat.

Spojovaný režim reprezentuje klasický způsob přístupu k používání tohoto protokolu. Přenos dat je realizován uvnitř vytvořeného spojení, které je zabezpečeno proti ztrátě dat. Připadají v úvahu dvě níže uvedené možnosti otevírání připojení. Obě spadají pod architekturu klient-server.

- a) O připojení žádá centrální stanice. V tomto uspořádání centrální stanice zažádá o připojení, poté vyzve podružnou stanici k zaslání požadovaných dat a po jejich obdržení zažádá o ukončení spojení. V praxi se jedná celkem minimálně⁹ o 8 datových přenosů, z nichž jeden obsahuje žádost o zaslání dat, druhý žádaná data a ostatní zabezpečují spojení.
- b) O připojení žádá podružná stanice. Komunikace by probíhala v následujícím pořadí: podružná stanice zažádá v zadanou dobu o připojení, zašle datový řetězec a po potvrzení příjmu ukončí spojení. Jedná se minimálně o 6 datových přenosů. Ačkoliv je tento model efektivnější z hlediska množství přenášených dat, patrně by docházelo při větším množství podružných stanic k obsazení všech virtuálních kanálů v TNC a tím pádem k jeho zahlcení. Toto by znemožnilo dalším stanicím připojení.

Mezi hlavní výhody spojovaného režimu patří snadnost přenášení většího množství dat a spolehlivost. Naopak mezi nevýhody lze řadit velké množství redundantních přenášených dat. Taková se nepodílí na přenosu informace a slouží pouze k udržení spojení. Vzhledem k nízké kapacitě přenosového kanálu je možno obsloužit menší množství stanic. Tato metoda přístupu k problému je výhodnější v případech, kdy lze použít kvalitnější TNC, který má více komunikačních kanálů¹⁰, s nímž by bylo možno vytvořit zároveň spojení s každým měřeným místem. Tím odpadne nutnost neustálého připojování a odpojování.

Nespojovaný režim představuje alternativní přístup k problému. Po dobu svého působení v sítích packet-radio jsem se s daným způsobem komunikace zatím nesetkal. Jedná se o využití směrovaného broadcastingu na monitorovacím kanále, ačkoliv k tomuto účelu nebyl navržen. Od uvedeného přístupu k řešení problému, kdy se přenáší pouze datový řetězec, lze očekávat hlavně nízké zatížení kanálu, což umožní provozovat více podružných stanic zároveň.

⁹ Za předpokladu, že se některý z paketů může ztratit.

¹⁰ Existují TNC s až 220 virtuálními kanály.

Tuto myšlenku podporují následující skutečnosti. Samotný princip přenosu datového rámce protokolu AX.25 má dostatečně dobré zajištění proti vzniku kolizí a nezačíná vysílat, pokud je jakákoliv šance, že snímaný signál obsahuje náznaky modulace. Výhodou je maximální možné zjednodušení ovládání TNC ze strany PLC (Programmable Logical Computer), které by se omezilo pouze na sestavení a odeslání vysílacího řetězce ve vhodném tvaru. Zjednodušením celé komunikace se vytvoří prostor pro častější odesílání informací.

Nevýhoda tohoto přístupu je v možné ztrátě přenášených dat, která však díky možnosti častějšího přenosu, a tím pádem většího množství přenášené informace, nemusí představovat zásadní problém. Dále je zde omezení maximální velikosti přenášené datové části paketu dané protokolem AX.25 na 255 bytů. Vzhledem k režimu přenosu a snaze o zajištění minimálních ztrátovosti dat při přenosu je vhodné, aby byl paket co nejkratší¹¹. U krátkých paketů je pravděpodobnost přeslechu a jeho ztráty snížena na minimum. Poslední nevýhodou zmiňovaného přístupu je nutnost přizpůsobit programové vybavení pro tento režim. Práce centrální stanice v monitorovacím režimu a rozlišování struktury paketů vyžaduje sofistikovanější algoritmy, které je potřeba integrovat přímo do programu pro obsluhu TNC. To vše zvyšuje zatížení CPU centrální stanice a klade větší nároky na programátora při vývoji tohoto řešení.

Důležité je zmínit netechnický omezující faktor obsažený v legislativním prostředí ČR¹². Při komerčním použití datových přenosů platí na většině sdílených pásmech omezující poměr mezi časem pro vysílání a příjmem, který je stanoven maximálně na 1:20. Pokud si tento poměr promítneme do jednotlivých výše zmíněných způsobů přenosu dat, zjistíme, že pro spojovaný režim je nejvíce zatíženým uzlem centrální stanice, u které by byl dosažený poměr mezi vysíláním a příjmem pravděpodobně v rozmezí 1:2 až 1:4. Umělé snižování tohoto poměru by vedlo k dalšímu snížení již tak malé propustnosti sítě.

¹¹ Vzhledem k osobním zkušenostem z provozu v tomto pásmu považuji za vhodné nepřekračovat délku 80 znaků.

¹² Podmínky pro používání frekvenčních pásem nejsou upravovány přímo zákonem, ale Českým telekomunikačním úřadem, který vydává Generální povolení, jejichž důležitost lze srovnat s vyhláškou ministerstva.

2.6. Analýza možností využití TNC pro řízení podružných stanic

Pro řízení podružných stanic je nutné zajistit obousměrné spolehlivé doručení datových rámců. Vzhledem k charakteru nespojovaného režimu připadá v úvahu pouze plnohodnotný spojovaný režim včetně všech jeho výhod a nevýhod. Po realizaci spojení se z virtuálního datového kanálu stává bitová trubka, která obousměrně doručí zaslaná data. Lze tak vzdáleně řídit stanici v rozsahu, který jí umožní některý řídicí komunikační protokol. Jediné omezení vyplývá z případné fragmentace balíku dat větší jak 255 bytů, kde dojde k rozdělení přenášené sekvence bytů na více paketů v jednom datovém rámci.

Důležitý a mnohdy i rozhodující faktor v oblasti řízení je dopravní zpoždění, od něhož můžeme odvodit vhodnost či nevhodnost nasazení do určitého typu technologie. V síti packet-radia trvá kompletní vyřízení jednoho paketu včetně potvrzení řádově sekundy. Za dobrých podmínek a slabého provozu se jedná zhruba o 1-2 sekundy. V případě silného provozu se časová prodleva zvýší k 15 a více sekundám. Vzhledem k uvedeným vlastnostem připadá v úvahu nasazení u technologií s dlouhou časovou konstantou, nebo u technologií, kde nezáleží na době odezvy. Případná aplikovatelnost této technologie v průmyslu je tedy omezená na několik málo typů řízených technologií. Příkladem průmyslového využití může být řízení výměňkových jednotek a energetických rozvaděčů.

3. Návrh řešení sítě pro bezdrátový sběr dat

Pro lepší představu o praktických požadavcích na strukturální a funkční vlastnosti bezdrátové sítě jsem se rozhodl seznámit se lépe s jedním konkrétním druhem technologie, u které by bylo teoreticky možné očekávat případné praktické nasazení, a posléze celý návrh přiblížit jejím reálným požadavkům. V tomto ohledu mi vyšla vstříc firma Tepelné hospodářství města Ústí nad Labem, která zajišťuje rozvody tepla a teplé užitkové vody, proto celá koncepce sítě předpokládá vzdálené řízení a sběr dat z výměníkůvých stanic.

3.1. Stručná charakteristika řízené technologie

Výměniková stanice je zařízení, které zajišťuje přestup energie, konkrétně tepelné. Vstupním médiem je pára nebo horká voda a výstupními médii jsou teplá užitková voda (dále jen TUV) a topná voda (TV). K přestupu tepla dochází v tepelném výměníku, což je nádoba, v níž jsou umístěny tenké trubky. Uvnitř trubek prochází topné médium ohřívající přestupem energie vodu vyplňující zbytek obsahu nádoby. K zajištění distribuce TV a TUV spotřebitelům jsou použita oběhová čerpadla.

Celá regulace se dělí zpravidla na 3 samostatné okruhy - okruh páry, okruh TUV a okruh TV. Cílem regulace výměnikové soustavy je zajistit konstantní teplotu TUV a správnou teplotu TV na výstupu z výměnikové stanice a v případě poruchy odpojit okruh na němž se porucha objevila. Správná teplota TV se určuje podle topné křivky a je závislá na venkovní teplotě. Hlavním úkolem bezdrátové komunikace v systému výměnikové stanice je ohlásit na dispečink havarijní stav a umožnit dispečerovi vzdáleně odpojit, nebo připojit jednotlivý okruh. Měření a vizualizace stavových hodnot systému jsou v tomto případě úlohy spíše doplňkové a ne příliš důležité. Blokové schéma výměnikové stanice je uvedeno v příloze A. Tato výměniková stanice již spadá do kategorie větších a složitějších stanic. Mým úkolem je navrhnout celou koncepci sítě a datový formát rámce umožňující propojení této a dalších jí podobných stanic.

3.2. Základní návrh koncepce sítě

Na základě zjištěných informací jsem se rozhodl použít síť s centrálním uzlem, v němž se budou veškerá data vyhodnocovat a uchovávat. Síť musí umožňovat přenášení dat z podružných stanic v pravidelných intervalech a v datovém řetězci se musí vyskytovat informace o funkčnosti stanice, čase odeslání zprávy a volitelně také některé doplňkové informace, které mohou mít charakter analogové hodnoty, digitálního stavu a stavu čítače¹³, jejichž množství je závislé na konkrétních potřebách. V případě poruchy je nutné vytvořit mechanismus, který zajistí spolehlivé odeslání informace o vzniklém problému okamžitě po její detekci a nezávisle na době, kdy má být odeslána další relace. Doba mezi jednotlivými relacemi může být relativně dlouhá, a to v řádu desítek minut. Je vhodné, aby v případě, že je z centrály prováděno řízení, se ostatní stanice odmlčely a uvolnily celý prostor pro potřeby řízení. Pokračování v podávání hlášení z podružných stanic by mohlo následovat po několika minutách po ukončení přímé komunikace mezi centrálou a podružnou stanicí. Centrální uzel musí umožňovat v případě potřeby vytvořit spojení s podružným uzlem a v omezených mezích mít možnost jej vzdáleně ovládat. Je vhodné pro řízení podružných stanic navrhnout takový protokol, aby jeho implementace v PLC byla co nejjednodušší.

Při vlastní realizaci je nutné centrální stanici obsluhující TNC umístit blízko k anténě, aby se zamezilo útlumu na spojovacím kabelu a poklesu kvality signálu. Proto jsem se rozhodl programovou část rozdělit na část zajišťující obsluhu TNC a archivaci dat a část zabezpečující vizualizaci uložených dat a vzdálenou obsluhu podružných stanic. Obě části spolu komunikují prostřednictvím jednoduchého protokolu nad TCP/IP v rámci počítačové sítě. Takové uspořádání rozšiřuje možnosti sledování stavu jednotlivých podružných stanic i pro všechny další oprávněné uživatele, čímž se celý systém stává flexibilnější. Uspořádání také umožňuje v případech, kdy je složité navázat přímou bezdrátovou komunikaci mezi dvěma odlehlejšími oblastmi, vytvořit hybridní síť a část informace přenášet počítačovou sítí TCP/IP.

Monitorovaná data se budou přenášet v nespojovaném režimu. Hlavním důvodem k tomuto kroku je snaha o zjištění skutečných vlastností této experimentální metody a její vhodnost či nevhodnost k reálnému nasazení.

¹³ Tento typ přenášené hodnoty byl uveden v zadání diplomové práce. Na zařízeních THMU nejsou měřeny hodnoty, které by vyžadovaly tento typ přenosu.

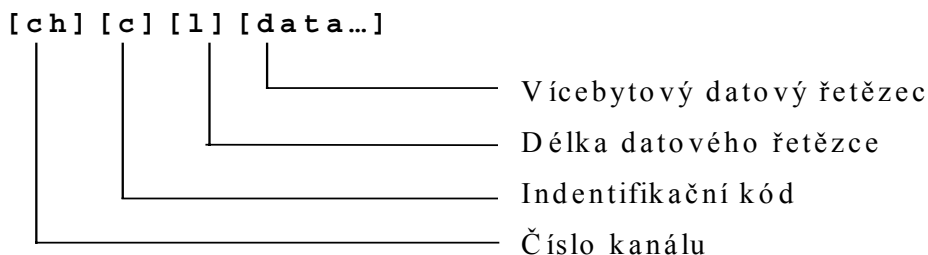
3.3. Návrh metod pro implementaci v programovém vybavení

Před vlastní realizací programového vybavení je vhodné nastítnit základní mechanismy a metody, které budou posléze implementovány do jednotlivých aplikací. Stěžejním a v mnoha ohledech nejnáročnějším prvkem celé aplikace je program obsluhy TNC a archivace dat, proto je zapotřebí provést dílčí analýzu jednotlivých funkčních bloků.

3.3.1. Metoda obsluhy TNC

Použité TNC podporuje ovládání ve dvou režimech. Jsou to HOSTMODE a Terminal-Mode. Jak již bylo zmíněno v kapitole 2.4. v části o používaných protokolech, režim Terminal-Mode není vhodný pro algoritmizaci. Před vlastní implementací programu pro obsluhu TNC je velmi důležité seznámit se důkladně se strukturou komunikačního protokolu HOSTMODE. Popis následující části vychází ze zdrojů [3], [14], [15], [16], z nichž je potřeba vybrat nejdůležitější části a přizpůsobit je verzi firmware použité u TNC-2 firmy STABO.

Po zapnutí napájení, nebo po resetu se TNC nachází v režimu Terminal-Mode. Do režimu HOSTMODE se přepíná sekvencí <ESC>JHOST 1<CR>. Při zasílání dat do TNC se používá jednotný třibajtový formát hlavičky znázorněný na obrázku 5.



Obr. 5 Formát pro odesílání dat do TNC režimu HOSTMODE

Kanál představuje jednobytovou hodnotu mající význam čísla virtuálního kanálu. U daného TNC se používá pro komunikační kanál rozmezí „0x01h“ až „0x0Ah“ a hodnota „0x00h“ reprezentuje monitorovací kanál. Druhý byte označený „info/příkaz“ má hodnotou „0x00h“, jestliže se jedná o data k vysílání, a „0x01h“ pokud data budou obsahovat příkaz pro TNC. Třetí byte označený „délka“ reprezentuje délku datové části datagramu a minimální hodnota „0x00h“ představuje datový řetězec o délce 1 byte. Následující sekvence bytů označená jako „data...“ již obsahuje vlastní data určená k odeslání, nebo příkaz pro TNC.

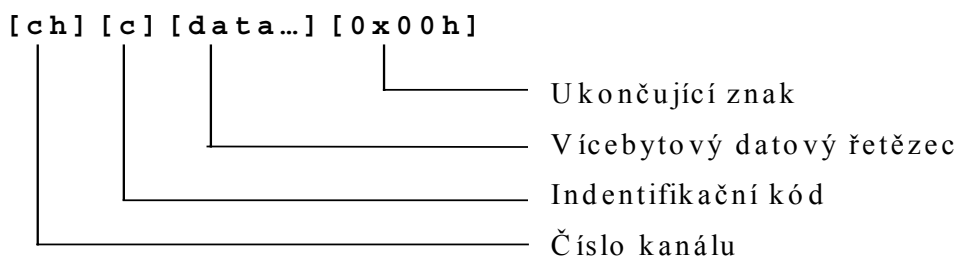
TNC používá tři různé formáty pro zaslání odpovědi počítači.

- a) **Krátký formát** zprávy reprezentuje potvrzení přijetí příkazu, nebo informaci, že je vše v naprostém pořádku. Složení ilustruje obr.6.



Obr. 6 Uspořádání krátkého formátu zprávy režimu HOSTMODE

- b) **Nulou ukončený formát** obdržíme od TNC ve tvaru, jehož struktura je patrná z obr.7.



Obr. 7 Uspořádání formátu ukončeným nulou v režimu HOSTMODE

Identifikační kód rámce se pohybuje v rozmezí čísel hodnot „0x01h“ až „0x05h“ Daný formát se používá pro různá stavová hlášení, jejichž význam lze rozdělit do následujících několika logických skupin.

Za kódem 1 následují řetězce obsahující odpovědi na dotazy nastavení TNC.

Za kódem 2 následují hlášení o chybách při komunikaci s TNC. Možné alternativy těchto hlášení jsou:

```

TNC BUSY - LINE IGNORED
INVALID COMMAND
CHANNEL ALREADY CONNECTED
STATION ALREADY CONNECTED

```

Kód 3 je použit u hlášení oznamujících změnu stavu linky na kanálu „ch“ k uživateli „volačka“ po trase „cesta“. Alternativy, které mohou nastat, jsou:

```

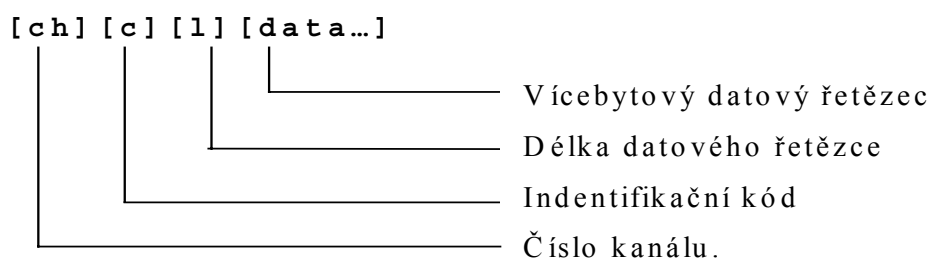
(ch) BUSY fm [volacka] via [cesta]
(ch) CONNECTED to [volacka] via [cesta]
(ch) LINK RESET fm [volacka] via [cesta]

```

```
(ch) DISCONNECTED fm [volacka] via [cesta]
(ch) LINK FAILURE fm [volacka] via [cesta]
(ch) FRAME REJECT fm [volacka] via [cesta]
CONNECT REQUEST fm [volacka] via [cesta]
```

Obsah zpráv u kódu 4 a 5 obsahuje informace o dění na pásmu získané prostřednictvím monitorovacího kanálu. Tyto zprávy budou klíčové pro identifikaci stanice u mnou zamýšleného nasazení nespojovaného režimu při sběru technologických dat. Jsou to informace o odesílateli, adresátovi, cestě, typu přenosu a pozici v datovém rámci.

c) **Formát s definovanou délkou** má hlavičku utvořenou následující sekvencí tří bytů.



Obr. 8 Uspořádání formátu s definovanou délkou v režimu HOSTMODE

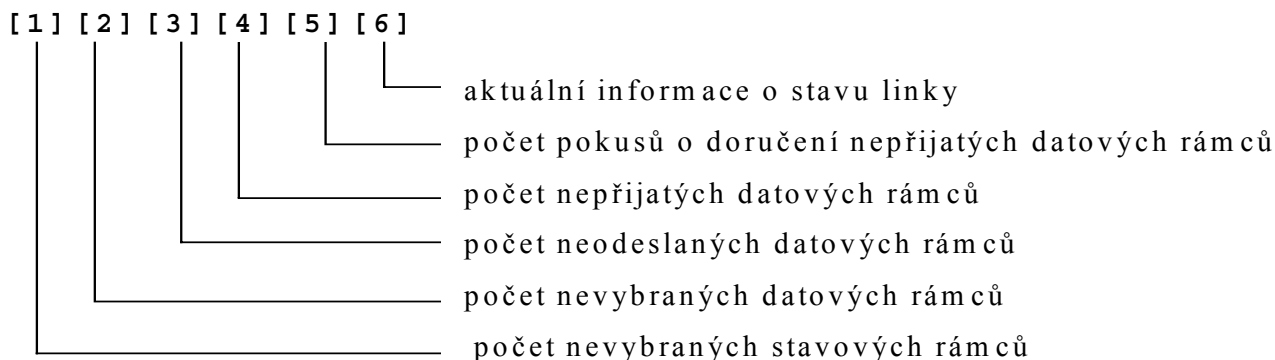
V tomto formátu obdržíme od TNC jednotlivé datové řetězce. Identifikační kód může nabývat hodnot 6 a 7. Zprávy obsahující kód 6 pocházejí výhradně z monitorovacího kanálu a zprávy obsahující kód 7 jsou z virtuálního kanálu uvedeného v prvním bytu.

3.3.2. Nejdůležitější ovládací příkazy režimu HOSTMODE

G „get“ – vyzvedne z požadovaného kanálu jeden datový rámec. V rozšířené specifikaci HOSTMODE od DG3DBI existuje speciální případ použití tohoto příkazu. Lze jej označit „G 255“ a funguje pouze v režimu HOSTMODE. Odpověď TNC na tento příkaz obsahuje kód, který určuje kanály obsahující nevyzvednutá data.

C „connect“ – jedná se o požadavek na připojení kanálu k požadované protistanici. Součástí příkazu může být kompletní cesta k cílové stanici, pokud její délka nepřesáhne osm převáděcích stanic. Nejjednodušší použití představuje příklad příkazu v konečném tvaru, v jakém se odesílá do TNC: „0x02h 0x01h 0x0Eh C CZ0NTU CZ0NLI 0x0Dh“.

L „*status-list*“ - příkaz z požadovaného kanálu vyzvedne informaci o stavu příslušného kanálu. Stavová informace je složená z šesti bytů, jejichž význam zle vyčíst z obr.9.



Obr. 9 Významový popis jednotlivých částí odezvy na příkaz „L“ ze sady *TheFirmware*

D „*disconnect*“ – tímto příkazem žádáme TNC o odpojení příslušného kanálu nezávisle na stavu komunikace, která na něm probíhá. Přátelskou formu ukončení komunikace, u níž obě strany očekávají její konec, je potřeba řešit na úrovni protokolu řízení podružné stanice.

3.3.3. Důležité konfigurační příkazy TNC

Kromě příkazů obsluhy linky obsahuje protokol HOSTMODE také příkazy pro konfiguraci jednotlivých parametrů TNC. Většina z nich je navržena tak, aby při volání bez parametru TNC udaly aktuální nastavenou hodnotu a při volání s parametrem se nastavila nová hodnota.

I „*callsign*“ - tímto příkazem se nastavuje volací znak celého TNC nebo jen jednotlivého kanálu. Tento řetězec je nejdůležitějším znakem určujícím identitu zařízení v síti.

T „*tx-delay*“ – tento příkaz nastavuje délku signálu otevírajícího komunikační kanál. Standardní délka se pohybuje v rozmezí 25 až 35 milisekund. Krátká doba nedokáže spolehlivě otevřít komunikační kanál, zatímco dlouhá doba snižuje efektivní propustnost sítě.

M „*monitor*“ – tímto příkazem definujeme vlastnosti chování, respektive paketový filtr monitorovacího kanálu. Jednotlivými volbami povolíme nebo zakážeme ukládání

různých částí jednotlivých paketů, které se vyskytly na monitorovaném kanále, a jejich pozdější přenášení do nadřazeného systému. Jak již název napovídá, monitorovací režim byl původně určen pouze ke sledování ostatní komunikace na pásmu a na jejím základě šlo určit přítomnost jednotlivých účastníků provozu. Nastavení s parametrem „N“ zakazuje používání monitorovacího kanálu a parametr „C“ pro změnu povoluje veškeré složky komunikace. Pokud chceme sledovat jenom některé složky komunikace, můžeme povolit unproto rámce parametrem „U“, stavové rámce parametrem „S“ a datové rámce parametrem „I“. Jednotlivé režimy lze navzájem kombinovat.

QRES „*quick reset*“ - tento příkaz kompletně resetuje celé TNC a uvede jej do továrního nastavení. Po jeho aplikaci se TNC nachází v režimu Terminal-Mode a je potřeba nastavit alespoň volací znak.

JHOST „*hostmode*“ - příkaz přepíná mezi režimy Terminal-Mode a HOSTMODE. „JHOST0“ aktivuje režim Terminal-Mode a „JHOST1“ aktivuje režim HOSTMODE.

O „*maxframe*“ - tímto příkazem se nastavuje systémová proměnná udávající maximální počet paketů v jednom rámci. K vysílání datového rámce delšího jak jeden paket dochází, pokud potřebujeme přenést najednou řetězec delší než 256 znaků. Kromě fragmentace dat zvolené při zasílání dat do TNC uživatelem může nastat situace, kdy TNC přehodnotí přenosové možnosti linky a zkrátí používanou délku paketu na vhodnější délku. V případě, že k tomuto jevu dojde, nedochází z hlediska uživatele k žádné podstatnější změně kromě zpomalení přenosu, která je způsobená rozdělením původního jednoho paketu na víc kratších.

3.3.4. Metoda filtrování přijímaných paketů

Pokud u TNC zapneme monitorovací režimy U a I, získáme podrobný přehled o datových řetězcích, které se vyskytnou na sdíleném kanálu, a jejich majitelích. Centrální stanice bude monitorovat veškerou tuto komunikaci a prioritně zpracovávat část unproto rámce. Rozhodujícím parametrem je v tomto případě volací znak odesílatele, který poslouží jako klíč do databáze, v níž jsou veškeré monitorované stanice a jejich senzory. V případě, že se bude volací znak odesílatele shodovat s se stanicí, od níž očekáváme data, předá se celý datový rámec vyšší vrstvě, která provede jeho rozbor na jednotlivé elementy. Pokud datový rámec nebude odpovídat zvolenému formátu, bude zahozen. Takovým striktním kritériem

snadno zamezíme zdlouhavé analýze rámců, které mohou vzniknout při spojovém režimu, nebo jako produkt opakování paketu, který patří ke komunikaci některého dalšího možného účastníka provozu na pásmu.

3.3.5. Metody použité při ukládání dat

Pro ukládání dat je vhodné použít databázi. Hlavní důvody k tomuto kroku jsou rozsáhlá podpora různých databázových systémů pod většinou programovacích jazyků a skutečnost, že jejich vývoj a vlastnosti jsou dostatečně pokročilé, což zajišťuje vysokou efektivitu zpracování dat a také maximální flexibilitu pro případné připojení jiných systémů pro vizualizaci nebo vyhodnocování dat. V dnešní době je již dostatek vysoce kvalitních databázových systémů, které jsou zdarma. Při realizaci jsem se rozhodl použít databázový systém MySQL, ke kterému lze přistupovat mimo jiné také přes datové zdroje ODBC, které jsou integrované v systémech MS-Windows počínaje verzí Windows98. Důvody volby tohoto databázového systému jsou rozepsány v kapitole 4.2.

3.3.6. Metoda detekce havarijních stavů

V průběhu filtrování paketu obdržíme informaci o odesilateli, na jejímž základě z databáze získáme informaci o všech přidělených senzorech. Při vlastním rozboru paketu na jednotlivé složky je vhodné provádět první kontrolu případného vzniku havarijního stavu, která umožní eventuální automatické spuštění programu ošetřujícího havarijní stav nezávisle na funkčnosti vizualizačního programu. Kritéria pro detekci havarijního stavu lze rozdělit podle charakteru veličin.

Analogový senzor má v databázi uložený povolený rozsah hodnot, ve kterém by se měly přenášené hodnoty pohybovat. Klíčové informace pro určení správných hodnot jsou volací znak odesilatele a číslo analogového portu. Po uložení nové hodnoty do databáze se ukládaná hodnota porovná s povoleným rozsahem, a pokud se v něm nebude nacházet, vyhodnotí se jako havarijní stav, který spustí program informující obsluhu.

Digitální senzor má také v databázi uloženou informaci vymezující, kdy je detekován havarijní stav. Klíčovými parametry pro určení správných parametrů senzoru jsou volací znak odesilatele a pozice v paketu určující číslo portu. Vzhledem k tomu, že některé přenášené veličiny mohou být přípustné v obou případech, je možno u digitálního senzoru zakázat

detekci havarijních stavů. V daném případě se hodnota senzoru používá pouze pro vizualizaci stavu systému.

Podobný mechanismus detekce havarijního stavu je také nutné implementovat do programu pro vizualizaci hodnot.

3.3.7. Metoda předávání dat mezi obslužným a vizualizačním programem

Komunikace mezi programy pro vizualizaci a obsluhu TNC bude probíhat na úrovni síťového protokolu TCP/IP. Program obsluhující TNC bude mít otevřený jeden port, na který v občasných intervalech bude dostávat od vizualizačních klientů informaci o jejich otevřených portech a síťových adresách. Tyto se budou ukládat do „tabulky klientů“, a pokud bude obdržén nový paket z podružné stanice obsahující aktuální informace, tak podle tabulky klientů postupně rozešle všem nové informace. Pokud vyvstane potřeba některou ze stanic řídit, vizualizační klient zažádá obsluhu TNC o připojení společně s informací o novém portu, na kterém očekává odpověď. TNC otevře spojení na volném kanálu a veškeré vstupy a výstupy přesměruje na požadovaný port. Tato koncepce zajistí nezávislost přímého řízení na funkcích integrovaných v programu obsluhy TNC. Při ukončení komunikace se společně s uzavřením virtuálního kanálu zavře i TCP/IP. Je-li zapotřebí získat starší informace uložené v databázi, je možné přistupovat přímo do databáze, nebo vytvořit jednoduchý přístupový protokol pro komunikaci, který by byl postaven na SQL dotazu, a prakticky by se jednalo pouze o nepřímý přístup k databázi.

3.4. Návrh komunikačního protokolu

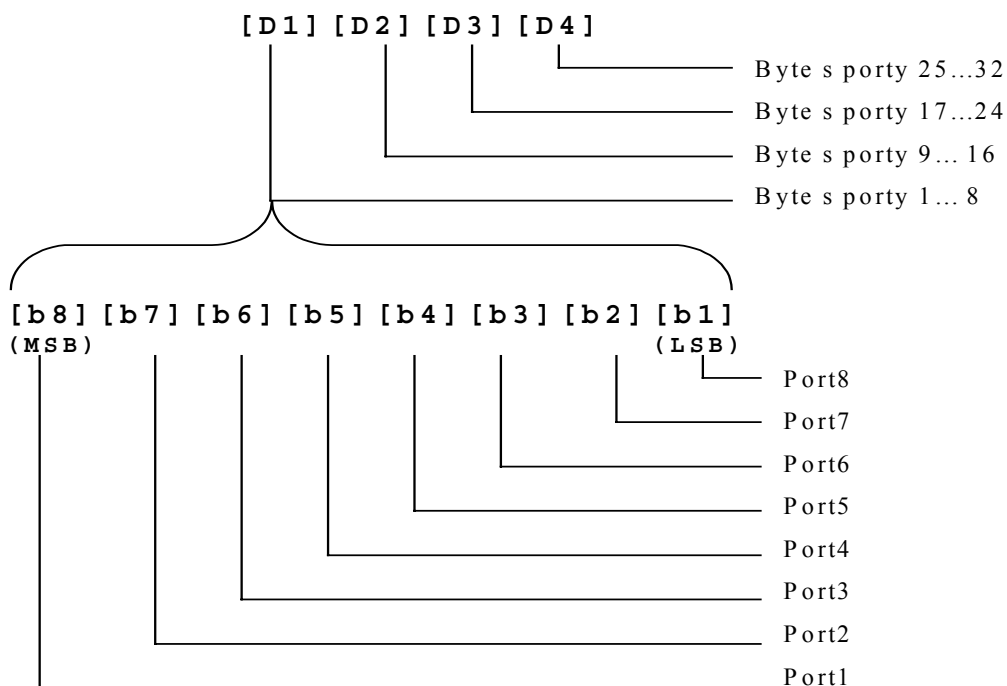
3.4.1. Návrh struktury datového rámce pro sběr dat

Při návrhu formátu zprávy datového rámce k přenášení měřených veličin mezi měřicím místem a centrální stanicí je potřeba zajistit několik základních vlastností. Výchozím požadavkem je jednoduchost tvorby datového rámce, aby v případě reálného nasazení, kdy bude TNC připojeno spíše přímo na RS232 rozhraní PLC než k osobnímu počítači s operačním systémem kompatibilním s win32, nepředstavoval implementační problém. Druhý požadavek na vlastnosti přenosového datového rámce spočívá v jeho zkrácení na minimální možnou „rozumnou“ délku, což pravděpodobně sníží ztrátovost paketů na únosnou míru a zároveň šetří kapacitu sítě pro další stanice. Ušetření přenosové kapacity a zároveň

zjednodušení protokolu lze realizovat přenášením dat v bytovém formátu a neomezovat se pouze na viditelnou část ASCII tabulky. Toto zjednodušení umožňuje velmi snadno vypisovat hodnoty stavů jednotlivých senzorů přímo ve formátu, v jakém se nacházejí v paměti bez potřeby další konverze.

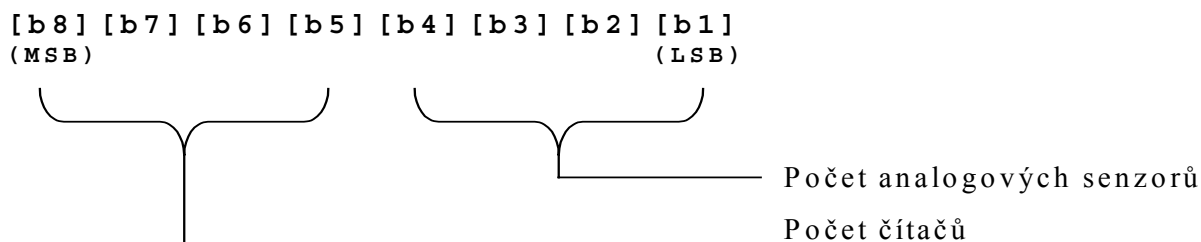
Nyní k jednotlivým přenášeným veličinám. Množství a důležitost jednotlivých typů senzorů jsem odvozoval z blokového schématu a tabulky senzorů stanice VS7, které jsou uvedeny v přílohách A a B. Vzhledem k požadavkům pouze na občasný přenos dat a zároveň na snadné určení data a času, kdy byl rámec vytvořen, jsem se rozhodl definovat nové a poměrně hrubé vzorkování času. Definoval jsem časovou jednotku o délce 5 minut a do datového formátu jsem času vyhradil tři byty. Při této délce je vytvořen prostor pro přesný popis data a času na příštích 140 let.

Druhou důležitou přenášenou veličinou je diskrétní stav vstupu či výstupu. Řídící systém výměňkové stanice VS7 má k dispozici 32 binárních vstupů a 32 binárních výstupů, z nichž je značná část použita pro řízení akčních členů v technologii a jejich informační hodnota pro dispečink je minimální. Kapacitně by mělo 32 přenášených diskrétních hodnot dostačovat. Pozici přenášející diskrétní hodnotu jsem označil jako „digitální port“, což je vhodný termín pro pozdější nasazení v aplikaci. Jednotlivými digitálními porty lze obsadit jednotlivé bity uložené za sebou do 4 bytů. Tyto hodnoty se vzhledem ke svému malému rozsahu budou přenášet vždy. Vlastní struktura je znázorněna na následujícím obrázku.



Obr. 10 Uspořádání části zprávy přenášející údaje o stavu digitálních portů

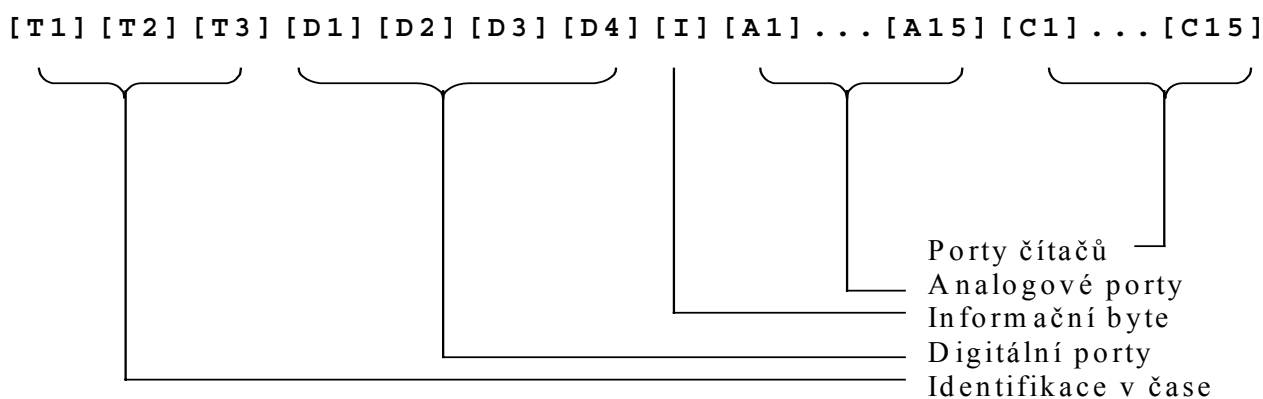
U přenášení analogových hodnot a stavů čítače je potřeba zvážit jejich množství, velikost a potřebnost při monitorování technologie. Z důvodů optimalizace délky datového rámce jsem se rozhodl umožnit dynamicky měnit délku rámce v závislosti na potřebném množství přenášených hodnot. K tomuto účelu přibyl do zprávy jeden informační byte, jehož funkce a struktura je patrná z obr. 11.



Obr. 11 Uspořádání informačního bytu definující délky částí zpráv

Při tomto uspořádání lze upřesnit množství přenášených hodnot na 0 až 15 analogových hodnot a 0 až 15 stavů čítače. Pro jednoznačné určení stavu analogové hodnoty jsou vyhrazeny 2 byty tvořící jedno 16bitové číslo. Uvedená dvojice bytů nesoucí

informaci o stavu analogové hodnoty nese pro zjednodušení označení „analogový port“. Vzhledem k tomu, že měřicí systémy nebývají osazovány 16bitovými převodníky, ale častěji verzemi s rozsahem 8 bitů, 10 bitů, nebo 12 bitů, bude se vyhrazený prostor obsazovat zprava do leva tak, aby se jeho hodnota neměnila. Nevyužitý prostor se doplní nulami. U čítačů lze považovat rozsah 16ti bitů za běžnou hodnotu, ale v případě, že bude rozsah menší, lze aplikovat stejná pravidla jako u zápisu analogové hodnoty. Také dvojici bytů vyhrazenou pro stav čítače je pro zjednodušení nazvána „port čítače“. Uspořádání jednotlivých prvků zprávy nejlépe vystihuje obr. 12.

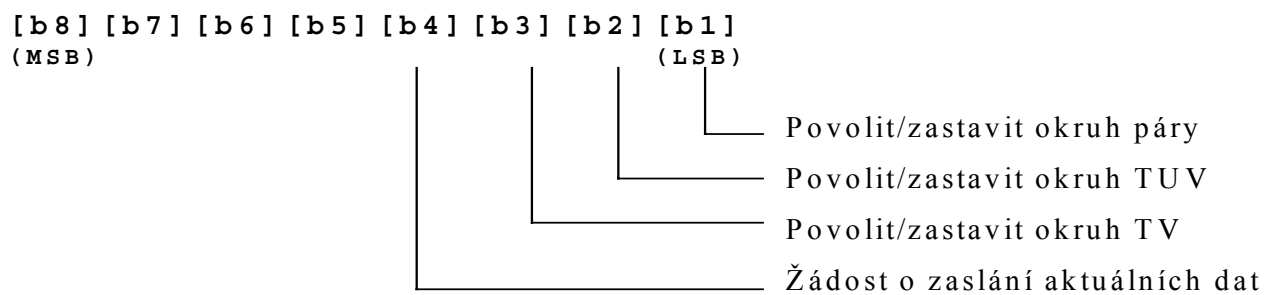


Obr. 12 Formát uspořádání jednotlivých částí zprávy

Minimální délka takto formulovaného formátu zprávy je 8 bytů a přenáší se informace o čase a stav všech digitálních portů. Informační byte v tomto případě ukončuje celou zprávu hodnotou 0x00h. Maximální možná délka zprávy je 68 bytů a přenášejí se informace ze všech typů portů.

3.4.2. Návrh protokolu pro řízení podružných stanic

Při vzdáleném řízení výměňkové stanice je potřeba zajistit možnost odstavení některé části provozu. Druhým úkolem řízení, který připadá v úvahu, je vytvořit mechanismus zjišťování dat aktuálního stavu systému. Při realizaci obou úkolů si vystačíme s přenášením jediného bytu. Významová struktura jednotlivých bitů je znázorněna na obr. 13. Logická jednička v tomto uspořádání reprezentuje požadavek na vykonání dané akce. Vzhledem k otevřené možnosti zasílání libovolného řetězce dat je tato část snadno modifikovatelná pro konkrétní aplikaci. Struktura založená na binárním stavu je výrazně jednodušší při implementaci do PLC oproti slovním formulacím používaným běžně v síti Internet.

**Obr. 13** Struktura řídicího bytu

4. Realizace programového vybavení

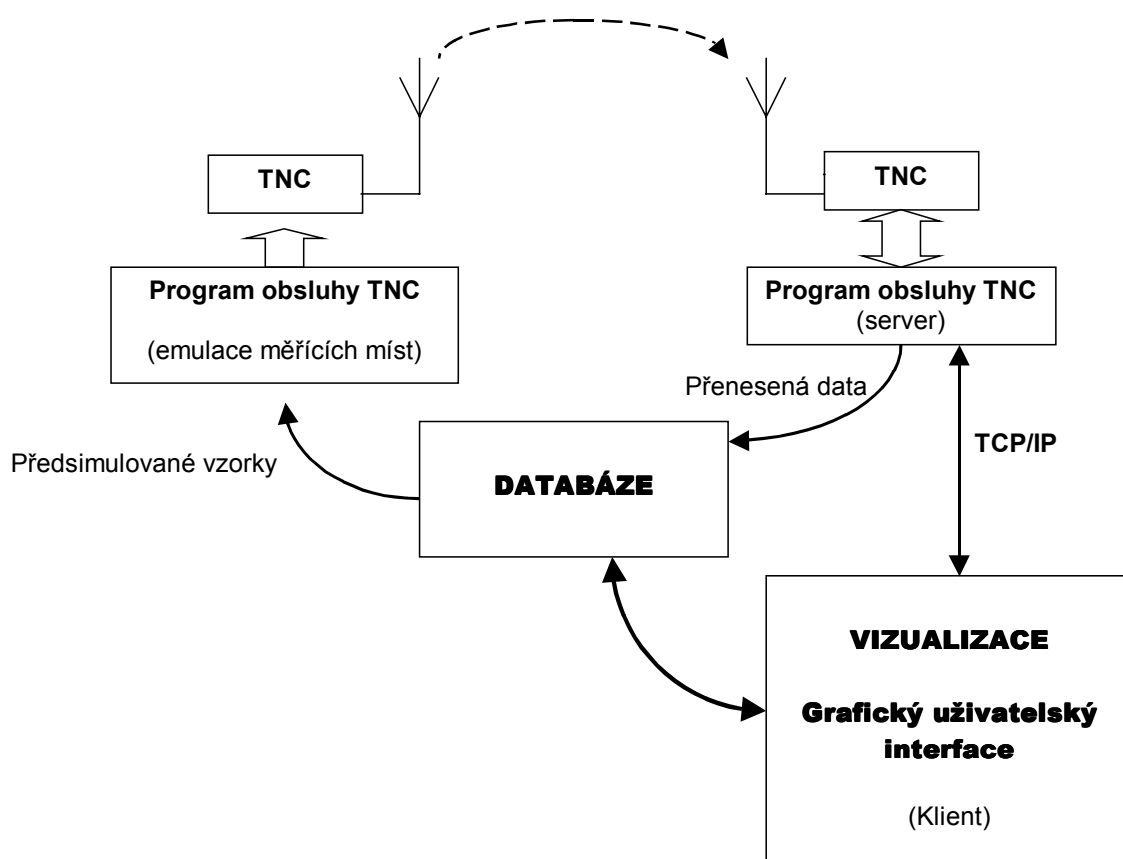
Pro ověření možného využití TNC k sběru dat v průmyslovém prostředí bylo potřeba vytvořit softwarové prostředky zajišťující provoz celého systému. Tento úkol představuje především sled operací potřebných k přenosu dat z jednoho TNC na druhé a následnou archivaci přenášených dat v databázi. Z uživatelského hlediska bylo žádoucí implementovat program převádějící přenášené hodnoty do vizuální podoby. Vzhledem k experimentální formě využití TNC byly také integrovány do programů prostředky pro následnou analýzu chování bezdrátové sítě.

Veškeré programové vybavení bylo vytvořeno ve vývojovém prostředí Borland DELPHI. Většina použitých komponent je přímo součástí DEPLHI a některé byly vytvořeny ze základních komponent. Jedinou cizí komponentou, kterou jsem použil, byla volně šiřitelná komponenta pro obsluhu sériového portu *QCCom32*, jíž poskytl internetový zdroj [17]. Jelikož značná část obsluhy zařízení TNC představuje práci s řetězcí, stala se nejdůležitějším nástrojem pro celé programování unita s funkcemi pro zpracování textových řetězců. Nazývá se *StringOP.pas* a obsahuje základní funkce pro parsování, porovnávání a nahrazování částí textového řetězce a transformacemi mezi jednotlivými použitými datovými formáty.

Vzhledem k tomu, že při realizaci předložené diplomové práce nebyla k dispozici reálná soustava výměňkové stanice s možností měření dat a zároveň byly k dispozici pouze dvě CB stanice vybavené modemy typu TNC, bylo nutné fyzicky upravit původně zamýšlenou koncepci do takové formy, aby bylo možno alespoň logicky realizovat celý projekt. Přenášená data bylo nutné vytvořit simulačním výpočtem a na jedné stanici s TNC emulovat více stanic. Druhé TNC bylo ponecháno v původně zamýšleném režimu centrální stanice. Celková koncepce realizovaného systému je lépe patrná z obrázku 14 a realizaci jsem rozdělil do tří programových celků – emulace měřících míst, sběr a archivace dat, vizualizace, z nichž každý má svůj specifický úkol.

Program obsluhy TNC „emulace měřících míst“ zajišťuje vybírání předsimulovaných dat z databáze a jejich vyslání do éteru. Základem tohoto programu je „Rozhraní pro správu a komunikaci s TNC“, detailněji rozebrané v kapitole 4.1., rozšířené o procedury a funkce potřebné k syntéze datového rámce.

Program obsluhy TNC „sběr a archivace dat“ také vychází z programu popsaném v kapitole 4.1 „Rozhraní pro správu a komunikaci s TNC“, je však rozšířen o implementaci mechanismů filtrování přijímaných paketů, jež byla podrobněji zmíněna v kapitole 3.3.4. zabývající se návrhem metod. Další rozšíření představuje implementace procedur a funkcí sloužících pro analýzu paketu, ukládání získané informace do databáze a serverovou část rozhraní pro komunikaci s vizualizačním klientem postavenou nad protokolem síťovým TCP/IP.



Obr. 14 Struktura jednotlivých celků při realizaci diplomové práce

Program vizualizace představuje klientskou část zabývající se zobrazováním stavů jednotlivých stanic, průběhů naměřených hodnot a hlášením případných havarijních stavů obsluze na dispečinku. Koncepce oddělení od programu obsluhy TNC byla zvolena z ryze praktických důvodů. Jedním z nich je nesporné zvýšení užitné hodnoty formou možnosti spouštění programu na více místech a zároveň možné efektivnější rozdělení činností, které souvisejí se správou celého systému. Druhý důvod k tomuto kroku vznikl v průběhu implementace „rozhraní pro správu a komunikaci s TNC“ a souvisí s objevenými chybami Win32 API, jež způsobují nekorektní a takřka stochastické chování systému.

Jednotlivým programovým celkům se diplomová práce dále věnuje v následujících podkapitolách.

4.1. Rozhraní pro správu a komunikaci s TNC

Implementace *rozhraní pro správu a komunikaci s TNC* probíhala na základě důkladné analýzy komunikačního protokolu HOSTMODE. Kromě vlastní tvorby procedur a funkcí pro syntézu a analýzu datových rámců, což reprezentuje jednodušší část problému, bylo potřeba sestavit takovou sekvenci příkazů a dotazů, aby se program neztratil, pracoval spolehlivě a z přichozího řetězce vyzískal maximum informace. Jelikož některé posloupnosti vyžadovaly několikerou komunikaci s TNC, která je oproti výkonu počítače relativně velmi malá, bylo nutné vybudovat speciální frontu dotazů pro TNC. Bylo zapotřebí zajistit, aby program věděl, která procedura má zpracovávat výsledek, a také odhadnout, jak dlouho může TNC zpracovávat daný dotaz.

Pro předávání hodnot jsem vytvořil některé nové datové typy. Paket reprezentuje typ pro element HOSTMODE datagramu v dekomponovaném stavu. *Tdotaz* je element pole, do kterého jednotlivé procedury zapisují pakety připravené k odeslání do TNC. V předložené diplomové práci bude toto pole nazýváno frontou dotazů. Datové typy jsou definovány následovně:

```
type paket=record
    chanel,code:byte;
    data:string[255];
end;
type Tdotaz=record
    pro,cekej:byte;
    paket_:paket;
end;
```

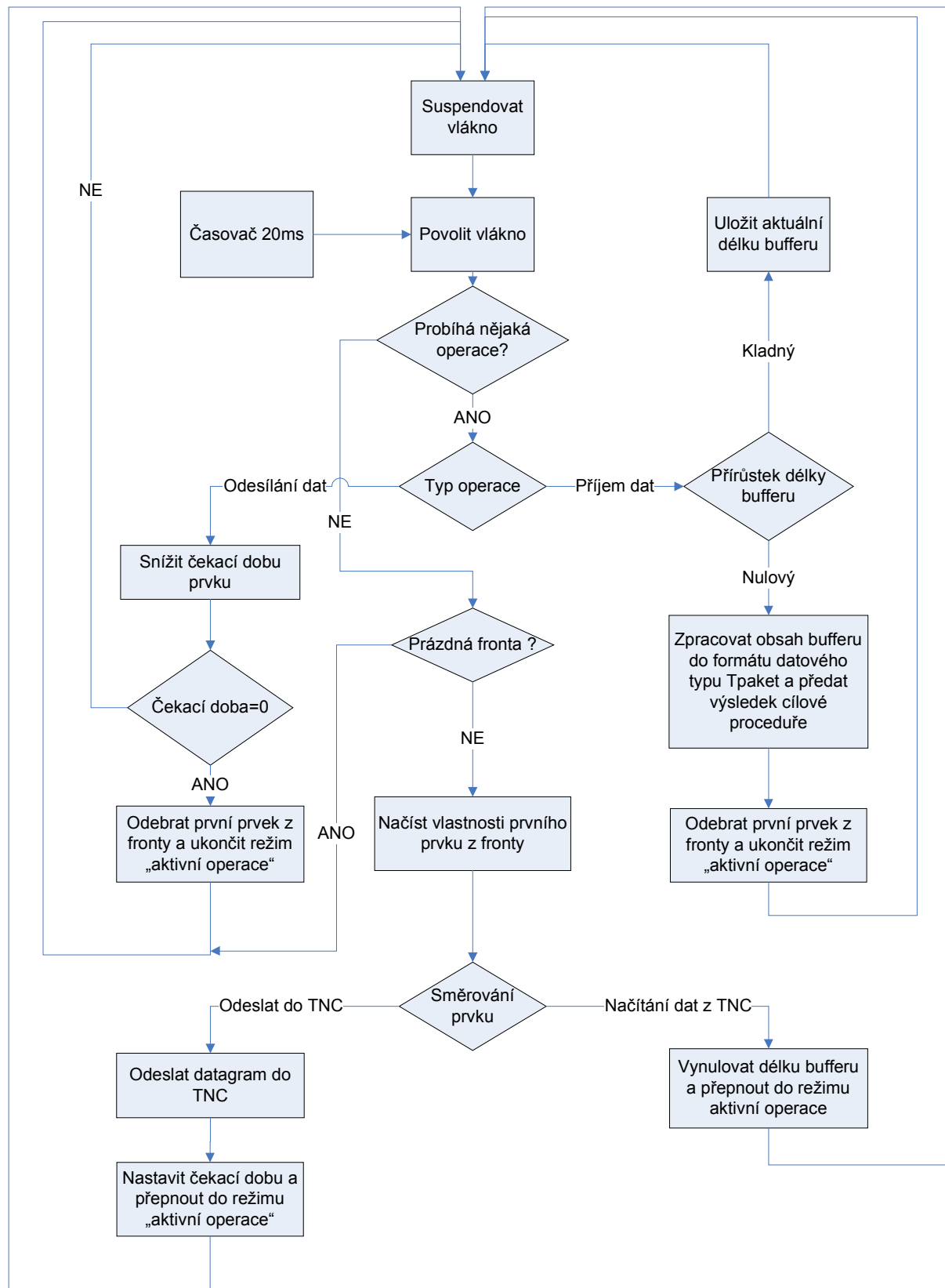
Fronta dotazů představuje typově FIFO frontu, proto pro její obsluhu byly definovány základní funkce zajišťující přidání prvku do fronty, získání prvního prvku, vyřazení prvního prvku a funkce zjišťující, zdali se ve frontě nachází nějaký prvek. Definice těchto funkcí jsou následující:

```
procedure AddToSeznam(dotaz:Tdotaz) ;  
function  GetFirstSeznam:Tdotaz;  
procedure GetOutFirstSeznam;  
function  SeznamEmpty:boolean;
```

Vzhledem k problémům spojeným s ovládáním sériového portu v reálném čase pod WIN32 API bylo potřeba optimalizovat obsluhu sériového portu tak, aby byla co nejefektivněji využita kapacita přenosového kanálu sériové linky RS232. Experimentálním způsobem jsem zjistil, že Win32 API aktualizuje obsah proměnné související se vstupním bufferem a sériového portu v pravidelných intervalech, které se pohybují v závislosti na nastavené prioritě vlákna v rozmezí 10 až 20 ms. Dále Win32 API nemá prostředky udávající počet neodeslaných bytů ve výstupním bufferu sériového portu.

Tyto vlastnosti způsobují, že se musí časy pro synchronizaci odesílání a přijímání datagramů odhadovat a kontrolovat průběžně příchozí data. Zpracování nekompletního datagramu vyvolává problémy se ztrátou části informace, což naruší celou řídicí sekvenci. Pro zamezení těchto vlivů bylo zapotřebí vyvinout algoritmus synchronizující celou komunikaci.

TNC komunikuje s počítačem rychlostí 9600 bps, což je přibližně 1 byte/ms. Čas potřebný k odeslání datagramu lze odhadnout z jeho délky, ale délka přijímaného datagramu je neznámá a může se pohybovat v rozmezí 2 až 259 bytů. V závislosti na době mezi aktualizacemi stavu Win32 API jsem stanovil jednotku času pro kontrolu množství bytu v příchozím bufferu na 20 ms, při níž lze očekávat, že celý datagram bude připraven k dalšímu zpracování za 1 až 13 časových kvant. Kritériem pro stanovení rozhodnutí o kompletnosti příchozího datagramu byla stanovena podmínka na nulovou derivaci funkce délky řetězce příchozího bufferu sériového portu. Směrování datagramu v rámci programu je určeno hodnotou entity typu *Tdotaz*, uložené na první pozici dotazové fronty. Jednotlivé procedury a funkce mají přidělena čísla, což umožňuje vytvořit sekvence příkazů a funkcí pro zpracování výsledků. Speciálním případem funkce je I/O rozhraní, které má přidělenou hodnotu „0“. Určení cíle a následníka je zajištěno duplicitou dotazů, z nichž první je určen vždy pro „0“ a druhý pro zpracující funkci.



Obr. 15 Blokové schéma algoritmu pro obsluhu sériové linky v reálném čase

Výhoda duplicity spočívá v možnosti zopakování dotazu, pokud dojde k chybě při komunikaci s TNC. Vlastní mechanismus synchronizace obsluhy sériové linky v reálném čase v podobě, v jaké byl zrealizován pro potřeby této diplomové práce, je zobrazen na obrázku 15.

Základní potřeby obsluhy TNC lze rozdělit na úlohy odesílání a příjmu datagramu. Při odesílání datagramu je použit jediný formát, proto je implementace této části jednoduchá. V programu je pro převod informace z vnitřního datového typu *paket* do řídicího řetězce připraveného k odeslání na sériový port použita funkce *CodeCommand*. Pokyn k odeslání datagramu do TNC je realizován přidáním dotazu do fronty dotazů. K tomuto účelu slouží funkce *StrToDotaz*, jež vnitřně rozšiřuje funkci *CodeCommand* o identifikaci následovníka. Definice výše zmíněných funkcí jsou následující:

```
function CodeCommand(text:string; chanel:byte; command:boolean):string;  
function StrToDotaz(data:string; chanel:byte; command:boolean;  
    pro:byte):Tdotaz;
```

Úloha čtení dat z TNC je výrazně složitější, ale lze ji pojmout jako konečnou sekvenci příkazů s případnými alternativami vedoucími k výsledku. Veškeré řetězce přijaté sériovou linkou zpracovává níže definovaná funkce *StrToPaket* a při dalším zpracovávání se mezi funkcemi předává informace v datovém typu *paket*.

```
function StrToPaket(text:string):paket;
```

Počátečním impulsem, který spouští mechanismus získávání informace z TNC, je vygenerování dotazu „G255“ časovačem. Výsledek se předá proceduře *G255*, která na základě předaného řetězce získá informaci o kanálech obsahujících nové informace. Pro jednotlivé kanály vygeneruje příkaz „L“, jehož výsledkem je informace o množství a charakteru datagramů, připravených k vyzvednutí z TNC. Jako následovníka určí proceduru *ListC*. Procedura *ListC* především rozebírá informaci o stavu virtuálního kanálu a výsledek ukládá do příslušné části pole *status*. Získáním této informace lze oddělit od sebe další obsluhu TNC na části zabývající se příjmem stavových informací a příjmem přenášené informace. Příjem stavové, nebo datové informace je realizován dotazem „G1“, nebo „G0“ na příslušný kanál. Kromě uvedeného rozdělení je zapotřebí informace o stavu linky zpracovat

v dalších kontrolních procedurách, které podle informace o počtu neodeslaných rámců a počtu nepřijatých rámců omezují požadavky na odesílání rámců, upravují kritéria délky paketu, a zabraňují tím zahlcení TNC a případnému pádu linky.

```
procedure G255(paket:paket);
procedure ListC(paket:paket);
type Tstatus=record
    lstatus,recived,unsend,uack,tryes,status:byte;
    call:string[50];
end;
status : array [0..10] of Tstatus;
```

Po vyzvednutí a rozložení datového řetězce, který přišel na virtuální kanál, se obsah zprávy uloží do malé vyrovnávací fronty na dobu nezbytně nutnou k vyřízení komunikace a předání obsahu vizualizačnímu klientovi. Funkce pro realizaci této komunikace jsou *DataSend* a *DataTransmit*. U stavových řetězců byly vytvořeny procedury a funkce zpracovávající nejdůležitější události, jimiž jsou připojení a odpojení kanálu a s tím související nucené odpojení klienta a udržování aktuálního stavu status - kanálové informace. Zpracování ostatních stavových informací není implementováno, protože pro ně nebylo nalezeno využití.

```
procedure connect_to(volacka:string);
procedure disconnect(mujkanal:byte);
function data_send(mujkanal:byte):paket;
procedure data_transmit(mujkanal:byte;text:string);
function get_status(mujkanal:byte):Tstatus;
procedure disc_client(mujkanal:byte);.
```

Při obsluze TNC bylo nutné zajistit některé doplňkové procedury a funkce, jejichž využití nalezneme v případě inicializace TNC při spuštění programu, nebo při vzniku neurčitého stavu a následné resynchronizaci ovládacího programu s TNC. O inicializaci TNC se stará procedura *TNC_config*. Jejím úkolem je nastavit parametry sériového portu, uvést TNC do režimu HOSTMODE¹⁴ a nastavit volací znak.

¹⁴ Nastavení režimu je realizováno zasláním dvou řetězců na sériový port níže uvedenými příkazy.
 qccom321.Write(#00+#01+#05+'JHOST0'); //vypnutí HOSTMODE
 qccom321.Write(#11+#17+#24+#27+'JHOST1'+#13); //zapnutí HOSTMODE

Po inicializaci je potřeba zjistit stav všech kanálů, což zajistí procedura *CallStatusList*. Tato procedura pouze zašle příkaz „L“ na všechny virtuální kanály TNC. Procedura pro zpracování výsledků této sekvence má jméno *StatusList* a kromě uložení aktuálních informací do pole status má za úkol požádat TNC o uvedení názvu připojené stanice na některém z virtuálních kanálů v případě, že jej nemá uloženu v poli status. U kanálů, na nichž bylo detekováno aktivní připojení, procedura *OpenKanal* zajistí nastavení vnitřních proměnných programu do takového stavu, aby byl kanál plně funkční, ukládaly se jeho data a bylo možno k němu přistupovat ze strany klienta. Definice použitých procedur jsou:

```
procedure CallStatusList;  
procedure StatusList(paket:paket);  
procedure OpenKanal(kanal:byte);  
procedure TNC_config;
```

Příjem na monitorovacím kanále je podroben filtraci na volací znaky. Pokud odpovídá volací znak na kanále získaný funkcí *detect_stanice* některé ze sledovaných stanic, je rámec podroben analýze typu. V případě že funkce *detect_type* vrátí hodnotu „1“ což reprezentuje I rámec a následující řetězec je datový rámec, je tento předán proceduře zabezpečující jeho dekompozici, kontrolu struktury a následné uložení do databáze.

```
function detect_type(s:string):byte;  
function detect_stanice(s:string):string;  
function zpracuj_data(data:string;stanice:string):byte;
```

4.1.1. Problémy vzniklé při implementaci

Jak jsem se již zmínil v úvodu realizace programového vybavení, při implementaci této části programu vznikly problémy související s nekorektním chováním a chybami uvnitř Win32 API. Chyby se týkají práce operačního systému se sériovým portem a vznikají v případě, kdy se předávají data systémovému FIFO bufferu sériového portu a zároveň vyprší časové kvantum přidělené k procesu. Při korektním jednání by mělo dojít k přerušení operace a opětovnému navázání ve chvíli, kdy je procesu opět přidělen procesor. Operační systém však při obnovení neodešle fyzicky celý datový rámec, ale pouze část. Systém však tuto operaci vyhodnotí jako úspěšně provedenou, tedy není možné přímo implementovat procedury pro řešení chyby, ale vzniklou chybu na základě různých příznaků odhadovat.

Tato chyba vzniká nezávisle na přidělené prioritě vlákna. Konzultace problému s podporou společnosti Microsoft vedla ke zjištění, že o problému vědí, ale vzhledem k malému množství lidí, kteří na tento problém poukazují, jej nehodlají řešit. Jimi doporučené řešení v podobě přenesení celé obsluhy sériového portu na druhý procesor však nepokládám za rozumné, neboť je nákladná a problém neodstraní.

Pro řešení problému jsem implementoval do programu různé testování příznaků, které většinu vzniklých chyb včas odhalí a problém vyřeší smazáním vstupního bufferu TNC a opětovným zasláním dat. Přes všechna opatření však není uvedený mechanismus naprosto spolehlivý. Může nastat sekvence nedetekovatelných chyb, která způsobí ztrátu synchronizace počítače s TNC.

4.2. Prostředky pro archivaci naměřených dat

Při implementaci programového vybavení byly v první fázi naměřená data ukládána do strukturovaných binárních souborů, ale s rostoucím objemem spravovaných dat se staly datové operace vyhledávání a třídění, zejména ty používané při vizualizaci, pomalé a na optimalizaci implementačně náročné. Řešení tohoto problému vedlo k provázáním aplikací s databázovým systémem (dále jen DBS). V rámci jednotného přístupu k databázi byla zvolena komunikace prostřednictvím datového zdroje ODBC. K vlastním operacím s databází jsem používal prostředky Borland Database Engine, převážně komponentu *tQuery*. Přestože prostředky jazyka SQL implementované v DBS odstranily implementační problémy práce s daty, i nadále přetrvával problém výkonové a časové náročnosti operací třídění dat pro vizualizaci. Ve snaze minimalizovat tento dopad a snížit čas potřebný k zpracování požadavku obsluhy byl optimalizován datový model, jehož výsledná podoba je v příloze C, a provedeny výkonové testy různých DBS.

Testoval jsem DBS Oracle, Firebird, MySQL, Paradox, Interbase, MS-SQL a Access 2000. Důraz byl kladen na rychlost odezvy SQL příkazu SELECT u velkých tabulek s řádově 10.000.000 záznamy. Při testech byly napřed vyřazeny DBS Paradox a Access, protože neumožnily vytvořit tak rozsáhlé tabulky. DBS Interbase a Firebird vykazovaly podobnou rychlost s ostatními u malého objemu dat, ale s rostoucím objemem se stávaly příliš pomalé. Vítězem testu byl DBS MySQL následován Oraclem.

Speciální kategorii představuje DBS MS-SQL, který výkonově lehce zaostával za systémem Oracle. Bohužel však vykazoval stochastické chování, pro něž jsem jej přestal považovat za solidní DBS. Problémy lze globálně popsat ztrátou integrity dat u velkých tabulek a projevovaly se rozdílnými výsledky při několikanásobném opakování téhož příkazu.

4.3. Vizualizační prostředí

Pro zobrazení přenášených informací v uživatelsky přijatelné formě bylo vytvořeno GUI (Graphic User Interface – grafické uživatelské rozhraní). Při jeho tvorbě se orientovaly snahy převážně na snadnost obsluhy, přehlednost a flexibilitu jak z hlediska rozšíření prostředí o další prvky, tak i pro nasazení na více počítačích zároveň. Celé prostředí lze rozdělit na „podkladovou plochu“ hlavní části vizualizačního prostředí, „inteligentní komponenty“ a část pro zobrazení grafů průběhů sledovaných veličin v požadovaném období.

4.3.1. Základní vizualizační komponenty

Jednotlivé vizualizační komponenty jsou koncipovány jako „inteligentní“. Hlavními úkoly těchto komponent je v požadovaném tvaru přehledně zobrazit hodnotu a detekovat havarijní stav. Implementace těchto prvků přímo do komponenty zpřehledňuje práci s nimi, šetří paměť a umožňuje je snadno rozšiřovat o další funkce nezávisle na zbytku systému.

V souladu se zadáním diplomové práce, kdy byla specifikována potřeba přenášet a zobrazovat stavy tří druhů hodnot – analogovou, digitální a stav čítače, vytvořil jsem pro každý typ přenášené hodnoty komponentu s vlastnostmi, které vhodně vystihují její charakter a jsou potřeba pro správné zobrazení dané veličiny. Všechny inteligentní komponenty mají některé vlastnosti podobné, což umožňuje jednotný způsob manipulace s nimi. Jsou jimi zejména parametry určující pozici na ploše, název signálu, přístupový port a mechanismus změny nastavení polohy komponenty a jejích jednotlivých parametrů. Tyto vlastnosti umožňují v konečném důsledku snadné a přehledné interaktivní nastavování komponent na ploše a snadnou tvorbu dalších ploch pro jiná monitorovaná stanoviště.

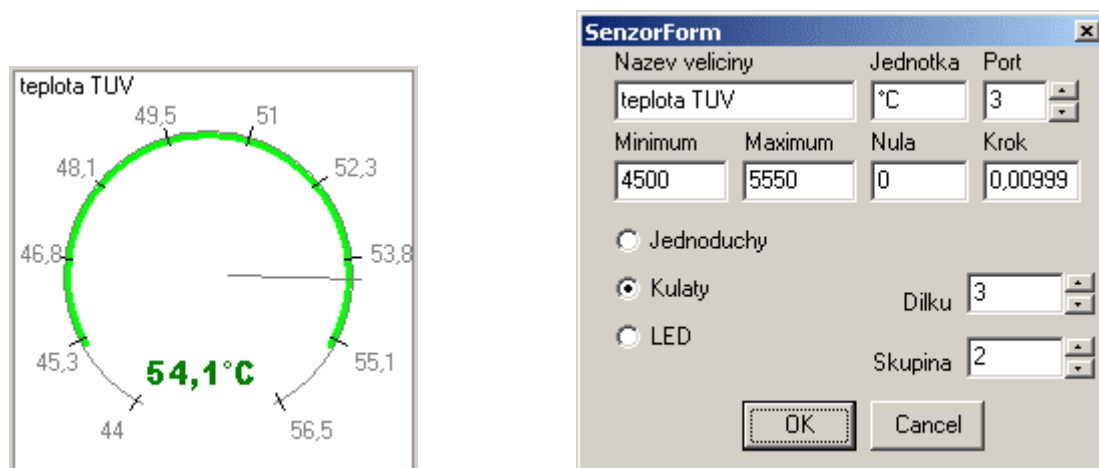
Jednotlivé rozšiřující možnosti jednotlivých komponent jsou přístupné přes jejich PopUpMenu. Možnosti jsou závislé na režimu, ve kterém komponentu provozujeme. Možné režimy jsou servisní a vizualizační. Volba režimu se nastavuje pomocí property *design*, a pokud je nastavená na hodnotu „true“, je komponenta v režimu servisním. V opačném

případě, tj. property *design* má hodnotu „false“, je komponenta ve vizualizačním režimu. Servisní režim se používá v části aplikace nazvané „nastavení senzorů“, jež je detailněji rozebrána dále v kapitole 4.3.3 „Jednoduchý editor pro rozmístění vizualizačních prvků na ploše“. Změna pozice jednotlivých objektů je možná pouze v servisním režimu. Aktivuje a deaktivuje se stiskem levého tlačítka myši.

Zdrojové kódy komponent jsou v příloženém kompaktním disku a nacházejí se v adresáři „Visual Component“.

4.3.1.1. Analogový senzor

Zobrazením analogového stavu se zabývá inteligentní komponenta *myAnalog*. Pro zobrazení hodnoty analogového senzoru jsem zvolil několik grafických interpretací. Jedna z nich představuje obdobu s běžným analogovým měřidlem s pohybující se ručičkou na pevné stupnici. Konkrétní podoba je patrná z levé části obrázku 16. Tato podoba je uživatelsky přirozená, a pokud není potřeba znát naprosto přesnou hodnotu, velmi dobře vystihuje pozici v rámci vymezeného rozsahu. Pro přehlednost je zobrazena také aktuální hodnota s veličinou a orientační název senzoru, ze kterého pochází uvedená hodnota. Aktuální hodnota se předává property *Value*, jejíž datový typ je word.



Obr. 16 Grafická podoba „analogové“ formy a servisní menu komponenty *myAnalog*

Základní nastavení senzoru je patrné z pravé části obrázku 16. Kromě tohoto servisního menu lze v konfiguračním režimu také pro každý senzor nastavit barvu pera v grafu a pozici na ploše.

Nyní podrobněji k jednotlivým položkám. Kolonka „Název veličiny“ a „Jednotka“ mají čistě informativní charakter. „Port“ představuje klíčový prvek pro identifikaci komponenty při získávání hodnot z databáze a je shodný s číslem „analogového portu“ definovaným v kapitole 3.4. při návrhu komunikačního protokolu.

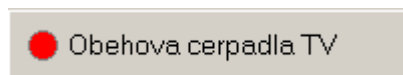
Kolonky „Minimum“, „Maximum“, „Nula“ a „Krok“ slouží pro přepočet hodnoty získané A/D převodníkem v PLC na skutečnou hodnotu a následně k detekci chybového stavu. Prozatím byla zvolena pouze lineární charakteristika senzoru na převodníku a převodní vztah odpovídá vyjádření přímky obecnou rovnicí ve tvaru $y = k \cdot x + q$, kde x, y, k, q jsou reálná čísla, k je směrnice, q je posun přímky na ose y .

Parametr „Skupina“ se uplatní až v momentě zobrazení průběhu veličiny v čase. Filosofie tohoto parametru vznikla z úvahy, že některá čidla mohou mít mezi sebou úzkou souvislost, a proto je vhodné je zobrazit do společného grafu. Příkladem je třeba teplota TV na vstupu a výstupu ze soustavy. Senzory mající stejnou hodnotu tohoto parametru jsou zobrazovány společně.

V případě, že dojde k překročení povoleného rozsahu, nastaví se property *chyba* na hodnotu „true“ a dojde k události *OnChyba*, kterou lze spustit libovolný kód v aplikaci.

4.3.1.2. Digitální senzor

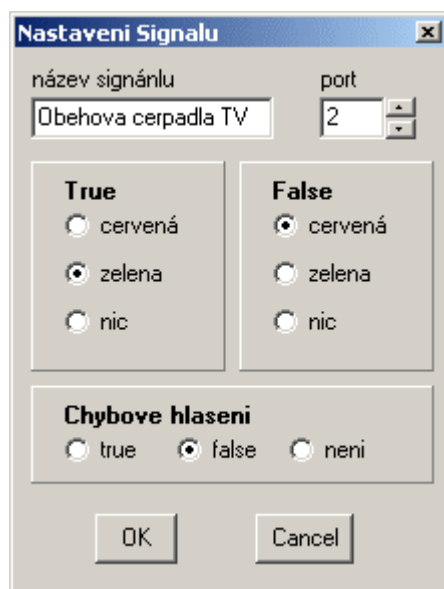
Komponenta *myDigita* slouží k zobrazení stavu digitálního portu. Spadá také do kategorie inteligentních komponent a je provedena pouze v jediné vizuální interpretaci představující dvoubarevnou LED diodu, která umožňuje graficky popsat tři základní stavy – svítí zeleně, svítí červeně a nesvítí. Realizovaná podoba objektu je patrná z obrázku 17. Aktuální hodnota se předává property *Value*, jež je datového typu word. Hodnota „0“ odpovídá stavu „false“ a jakákoliv vyšší hodnota představuje stav „true“.



Obr. 17 Grafická podoba komponenty myDigita

Nastavení komponenty v servisním režimu umožňuje nastavit základní identifikační prvky, jimiž jsou „název signálu“ a „port“, a položky související pro parametrizaci vizuálního

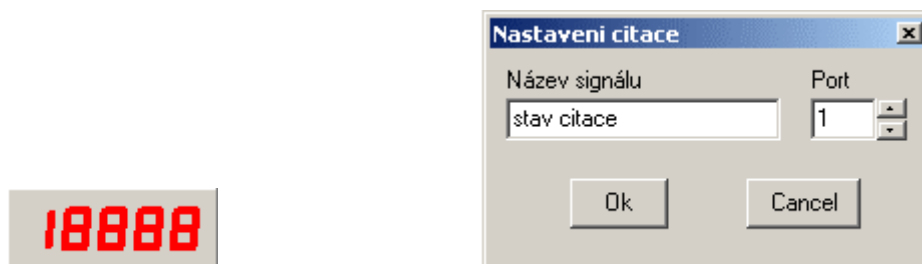
nastavení. Mezi tyto patří volby nastavení barvy pro možné stavy signálu a pravidlo pro vyhodnocení chybového stavu. V případě, že nastane chybový stav, tak se shodně se senzorem *myAnalog* vyvolá událost *OnChyba* a nastaví property *chyba* do stavu „true“. Konkrétní podoba konfiguračního menu pro nastavení jednotlivých vlastností této komponenty je na obrázku 18.



Obr. 18 Servisní menu komponenty *myDigital*

4.3.1.3. Stav čítače

K zobrazení hodnoty stavu čítače slouží komponenta *myCounter*. Při její tvorbě jsem neměl k dispozici informace týkající se konkrétního využití ve sledované technologii, proto jsem pro vizuální podobu zvolil podobu čísla zobrazeného na sedmsegmentovém displeji a k nastavení pouze nejnutnější údaje důležité pro jednoznačnou identifikaci komponenty nutnou k začlenění do jednotného systému. Komponenta přijímá hodnotu přes property *value*, jejíž datový typ je *word*. Konkrétní podoba komponenty a jejího konfiguračního menu je patrná z obrázku 19.



Obr. 19 Grafická podoba a servisní menu komponenty *myCounter*

4.3.1.4. Popisek

Pro zvýšení přehlednosti plochy se zobrazenými stavy jednotlivých senzorů a možnosti doplnění dalších informací textového charakteru, jimiž mohou být například adresa měřicího místa nebo místní slangový název objektu, jsem vytvořil komponentu *myLabel*. Vychází z komponenty *Label*, která byla rozšířena o nástroje pro interaktivní nastavování v prostředí. Z *PopUpMenu* jsou v editačním režimu přístupné položky „Editovat obsah“ a „Nastavit font“. Nastavování pozice je shodné s ostatními komponentami.

4.3.2. Hlavní část prostředí pro vizualizaci stavů podružných stanic

Úkolem této části programu je umožnit uživateli přepínat mezi jednotlivými stanicemi a různými režimy zobrazení požadovaných dat. Plocha je rozdělena na dvě části. V pravé části je přítomna tabulka se jmény monitorovacích stanic a v levé části rozsáhlá plocha pro zobrazování vybraných parametrů. Tuto lze přepínat mezi režimy: Aktuální stav, Grafy a Logování. Skutečná podoba je zobrazena v příloze D. Při přepnutí stanice dojde k načtení všech „inteligentních“ komponent včetně předání jejich parametrů a vytvoření jednotlivých exemplářů na podkladovou plochu listu „aktuální stav“. Jednotlivé komponenty jsou uloženy do pole komponent, což umožňuje je bez problémů kontrolovat a používat jejich vlastnosti.

Grafický list v nabídce obsahuje další listy pro výběr sledovaného typu senzoru. Jeden z nich obsahuje možnosti nastavení kritérií pro zobrazování grafů. Pro vykreslování grafů analogových hodnot se používá procedura *kresli*. Vstupní parametry představují startovní a koncový vzorek, který má být vykreslen, a ID stanice. Ostatní parametry si procedura získá od „inteligentních“ komponent a stavu položek na konfiguračním listu. Procedura rozdělí podle přednastavených skupin senzory, zkalibruje si maxima a minima a vykreslí je v požadovaném časovém intervalu. Procedura *kresli_digital* provádí obdobné operace u senzorů digitálních. Vzhledem k vysoké náročnosti provádění těchto operací na databázový systém se spouští až v momentě, kdy je zobrazení skutečně potřeba.

```
procedure kresli(start,konec,stanice:longint);  
procedure kresli_digital(start,konec,stanice:longint);
```

Pokud program obdrží nová data prostřednictvím TCP/IP klienta, zkontroluje je s parametry komponent uloženými v databázi. Pokud nastane chyba, vyvolá se procedura, která na vzniklý problém upozorní.

4.3.3. Jednoduchý editor pro rozmístění vizualizačních prvků po ploše

Tato část hlavního programu načítá komponenty a zobrazuje je na plochu podobně jako v případě „aktuálního stavu“. Rozdíl je v nastavení property *design* na hodnotu „true“. Komponenty jsou rozmístěné na ploše a jsou přístupné veškeré jejich konfigurační nástroje pro změnu vnitřních stavů. Pro uložení aktuálních hodnot do databáze proběhne procedura *UpdateDB*, která obnoví všechny záznamy do aktuálního stavu.

```
procedure UpdateDB;
```

4.4. Program simulace podružných stanic

Základem pro programovou část simulace podružných stanic je rozhraní pro obsluhu a správu TNC. Primárním úkolem je v pravidelných intervalech pod volacími znaky jednotlivých emulovaných stanic odesílat datové rámce obsahující nasimulovaná data. O správné načasování odeslání se stará vnitřní časovač, který kontroluje, zdali funkce *DateTimeToVzorek(now)* nevrátí novou hodnotu vzorku k odeslání. Vlastní sestavení datového řetězce zajišťuje funkce *VytvorPaket*, která data k odeslání vyzvedne z databáze prostřednictvím SQL dotazu a zakóduje je do formátu zprávy popsaneho v kapitole 4.3.

```
function DateTimeToVzorek(date: TDateTime): longint;  
function VytvorPaket(vzorek:longint;stanice:word):string;
```

4.4.1. Simulace hodnot veličin podružných stanic

Problematikou simulace veličin se zabývá pro tento účel speciálně vyvinutý program „Generátor“. Jedná se o velmi zjednodušený model výměňkové stanice, jehož výstupem jsou teploty TV a TUV na vstupu a výstupu ze soustavy a venkovní teplota. V uvedeném modelu nešlo o vytvoření relevantních dat metodou matematicko-fyzikální analýzy problému, ale spíše o vytvoření takových průběhů hodnot jaké očekávám na reálné soustavě výměňkové stanice.

Základem je několik paralelně jdoucích diferenčních rovnic s konstantními koeficienty, které byly zvoleny tak, aby výsledné chování vykazovalo nekmitavý průběh s jednotkovým zesílením a hlavní časovou konstantou vycházející zhruba na jednu hodinu. Příklad jedné z nich je uveden ve vztahu za tímto odstavcem. Zanesením malé náhodné odchylky do rovnic způsobily, že výsledné křivky působí realističtěji. Při simulaci digitálních hodnot byla do algoritmu vložena neurčitost, se kterou se vytváří anomálie od normální hodnoty pomocí náhodného generátoru. Kompletní zdrojový kód simulačního nástroje se nachází na příloženém CD v adresáři *generátor*. V adresáři *generátorSQL* je celý algoritmus rozšířený o SQL příkazy zajišťující uložení výsledku do databáze.

```
tout [0] := (-a [1] *tout [1] -a [2] *tout [6] -a [3] *tout [10]  
+b [1] *y [0] +b [2] *y [4] +b [3] *y [6] ) ;
```

4.4.2. Emulace více stanic na jednom TNC

Vzhledem k experimentální metodě využití protokolu AX.25 v nespojovaném režimu pro odesílání datových rámců bylo nutné vyvinout metody pro vytvoření zdání více stanic na odesílacím TNC pro spojovaný i nespojovaný režim. U spojovaného režimu bylo použito individuálních nastavení pro každý virtuální kanál příkazem „I“. Uvedená metoda v sobě neskýtá vážnějších problémů kromě omezení maximálního množství použitelných volacích znaků, což je dané množstvím virtuálních kanálů.

Při odesílání jednotlivých datových rámců prostřednictvím monitorovacího kanálu v nespojovaném režimu je nutné napřed nastavit volací znak monitorovacího kanálu. K tomuto účelu je opět vhodný příkaz „I“ určený pro kanál 0. Při odesílání tedy dochází k neustálému přenastavování vnitřních proměnných TNC vázaných ke kanálu 0. Dlouhodobější provoz v tomto režimu by však mohl způsobit poškození paměťových obvodů TNC, neboť pro takový provoz nebylo konstruováno.

5. Ověření funkčnosti a možnosti sítě

Již při implementaci jednotlivých funkčních celků byla důsledně testována jejich funkčnost a vývoj se ubíral cestou zajišťující splnění zamýšlené koncepce v plném rozsahu. Po celkové implementaci a ověření funkčnosti systému v krátkodobém testovacím provozu vyvstaly otázky spojené s chováním systému v prostředí intenzivního zatížení sítě a vlivů nastavení parametru délky paketu, popřípadě vliv okolního rušení na celkovou spolehlivost datového přenosu.

5.1. Vlastnosti testované sítě

Jak již bylo zmíněno v kapitole 2.5., snahou diplomové práce bylo především vyzkoušet možnosti aplikování v praxi neprozkoumaného přenášení dat v nespojovaného režimu, proto následující testy jsou orientovány převážně na otestování chování sítě v tomto režimu. Snaha o využití nespojovaného režimu byla motivována možností potvrzení domnělého zvýšení efektivity využití pásma za cenu částečné ztráty přenášené informace.

5.1.1. Použité vybavení a způsob testování chování systému

Pro testování sítě v reálných podmínkách byla využita stávající síť CB packet radia a pro vyšší objektivitu jednotlivých testů bylo testování provedeno na třech rozdílně dlouhých trasách, jež byly k dispozici. Oproti podmínkám odpovídajícím případnému komerčnímu nasazení systému, kdy je povolený maximální výstupní výkon 1W, používaly jednotlivé stanice vysílací výkon 4W. Bohužel použitá CB stanice neumožňuje snížit vysílací výkon, a proto nebylo možné podniknout kroky k zobjektivnění testů. Základní informace o jednotlivých účastnících testu jsou uloženy v tabulce 1. Vzdálenost byla zjištěna z lokátorů¹⁵ a vzhledem k systému jejich tvorby je zatížena značnou chybou pohybující se kolem $\pm 1,5$ km.

¹⁵ Globální systém lokátorů byl vytvořen radioamatéry pro přibližné určení polohy účastníka sítě. Skládá se ze šestice znaků a každý jednotlivý lokátor představuje oblast o rozměrech zhruba 2,7x2,7 km. Při určování uvedených vzdáleností byl použit, radioamatéry pro tento účel používaný, program QTHLOC.

volací znak	typ antény	lokátor	vzdálenost (km)	modem
CZ4KIO	5/8 3 radiály	JO70NU	-	TNC
CZ0NLI	1/2	JO70NT	4.6	Baycom
CZ4WAY	bumerang	JO70MS	11	Baycom
CB0SPB	neznámý	JO70IX	32.3	neznámý

Tabulka 1 Seznam a vybavení stanic používaných při testování

Vlastní testy byly prováděny metodou „relátkování paketu“ zpět k odesílateli s využitím funkce digipeater obsažené v protokolu AX.25. Cílem testů bylo zjistit poměr mezi ztracenými a neztracenými pakety. Na základě tohoto poměru lze posuzovat spolehlivost přenosu.

5.1.2. Závislost propustnosti sítě na přírodních podmínkách

Při ladění programů a následném testování sítě bylo pozorováno občasné odlišné chování, způsobující vyšší ztrátovost paketů. Při snaze tento jev objektivně popsat a definovat se vyskytl problém kvantifikovatelného popsání přírodních podmínek, které způsobují pozorované chování sítě takové, aby bylo možné vytvořit skupiny a naznačit alespoň řádově tendenci pozorovaného chování. Pro popis jsem zvolil veličinu, která nesouvisí s přírodními podmínkami, ale popisuje chování přenášeného signálu. Veličina se označuje „S“, měří se S-metrem¹⁶ a reprezentuje intenzitu signálu či šumu. Obecně lze tedy konstatovat, že existují dva základní vlivy snižující pravděpodobnost bezproblémového přenosu datového rámce. Jsou jimi útlum signálu a zvýšení úrovně šumu.

Útlum signálu bývá způsoben většinou vznikem objemových překážek, jakými jsou kupříkladu hustý déšť, mlha a sněžení. Útlum signálu v prostředí má oproti intenzitě rušení na kvalitu přenosu signálu výrazně menší vliv, avšak vytrvalý déšť či sněžení mohou způsobit další druhotný úbytek síly signálu rozladěním antény.

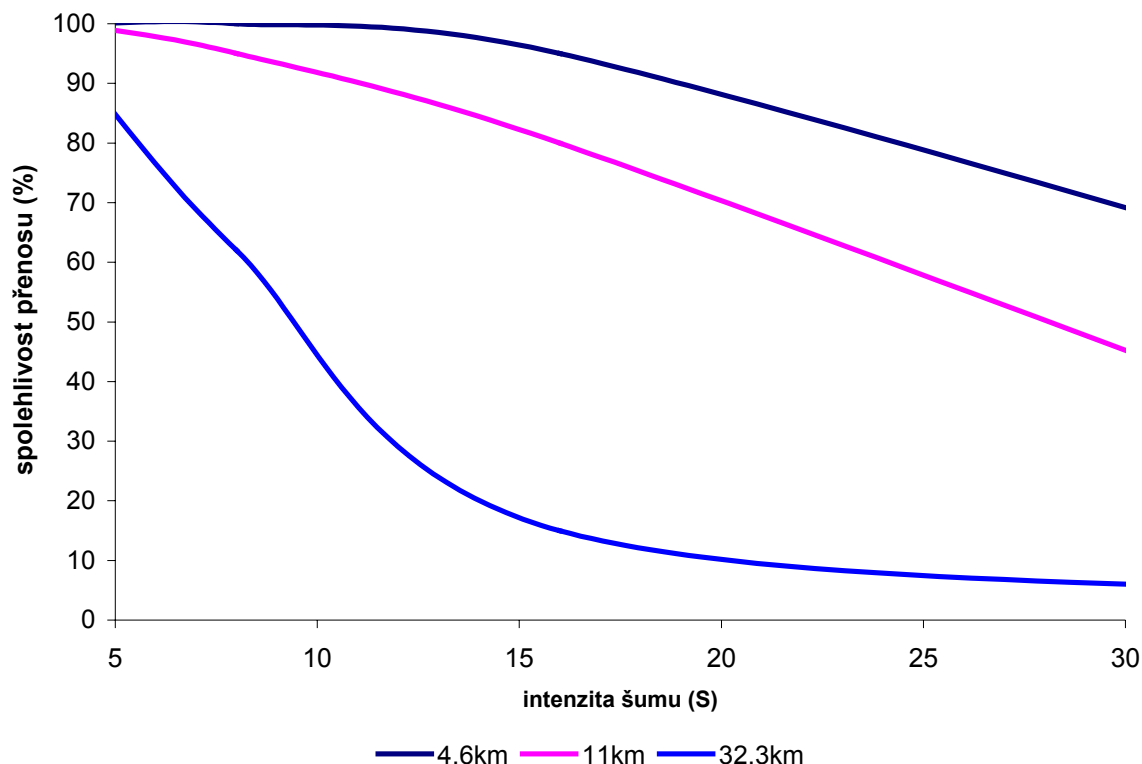
Okolní šum způsobují hlavně různé interference vlnění. Do výsledného šumu se mimo jiné projevuje vzdálený provoz na stejné frekvenci, ale nejvíce ovlivňujícím prvkem je sluneční záření, respektive množství slunečních skvrn. Na základě pozorování jsem zjistil, že nejvýrazněji se tento jev projevuje, pokud slunce svítí pod malým úhlem a obloha je bez

¹⁶ Pro měření byl použit S-metr vestavěný do CB stanice používající rozmezí hodnot 1 až 30. Výstupní veličinou je proud (mA). V praxi se používají různé stupnice. Nejčastěji lze nalézt rozmezí 1 až 10, nebo 1 až 30.

mraků. Jsou to tedy především východy a západy slunce, nebo značná část jasných dnů o zimních měsících. V dané situaci dochází k vytvoření jakéhosi „zrcadla“ v ionosféře způsobené slunečním zářením, které zapříčiní, že se od něj odráží radiové vlny. Výsledný šum je patrně tvořen vzájemně interferujícími signály z řádově větších vzdáleností, které by se bez vlivu tohoto slunečního efektu ztlumily. Popisovaný jev je velmi podobný lomu světla na přechodu dvou materiálů, kdy směr dalšího šíření záleží na úhlu dopadu elektromagnetického vlnění na ionosféru. U signálu z velké vzdálenosti je úhel dopadu malý a dochází k odrazu, zatímco signál z kratší vzdálenosti má velký úhel dopadu a prochází ionosférou do vesmíru. Uváděný mezní úhel je 41,6 stupně [4]. Uvedenou teorii lze podpořit skutečností, že v těchto situacích dochází často ke spojení na velmi dlouhé vzdálenosti. V průběhu vývoje programových prostředků pro předkládanou diplomovou práci jsem měl přímé spojení s Belgií, Finskem a Francií. Podrobněji se této problematice věnuje kapitola „Dosah radiostanic CB a šíření signálu o kmitočtu 27MHz“ obsažená ve zdroji [5].

Přibližný graf závislosti spolehlivosti přenosu na intenzitě okolního šumu je znázorněn na obrázku 20. Je nutné podotknout, že běžně dosahované hodnoty šumu se pohybují v rozmezí 5S až 8S, hodnoty 12S až 16S představují již vysokou hladinu šumu, kdy je pro člověka náročné rozeznávat fonickou komunikaci. Čistý signál ze vzdáleného zdroje mívá zpravidla sílu přesahující 20S.

Na základě zjištěných údajů lze hodnotit citlivost modemu typu TNC jako velmi dobrou, ba dokonce je datový typ přenosu výrazně odolnější proti rušení než běžná fonická komunikace. Očekávanou vysokou ztrátovost paketů při přenosu na velké vzdálenosti lze za normálních podmínek kompenzovat používáním spojovaného režimu, který zajistí spolehlivé doručení paketu. Při horších atmosférických podmínkách však roste ztrátovost do té míry, že i ve spojovaném režimu může docházet k častým výpadkům a nespolehlivosti spojení.

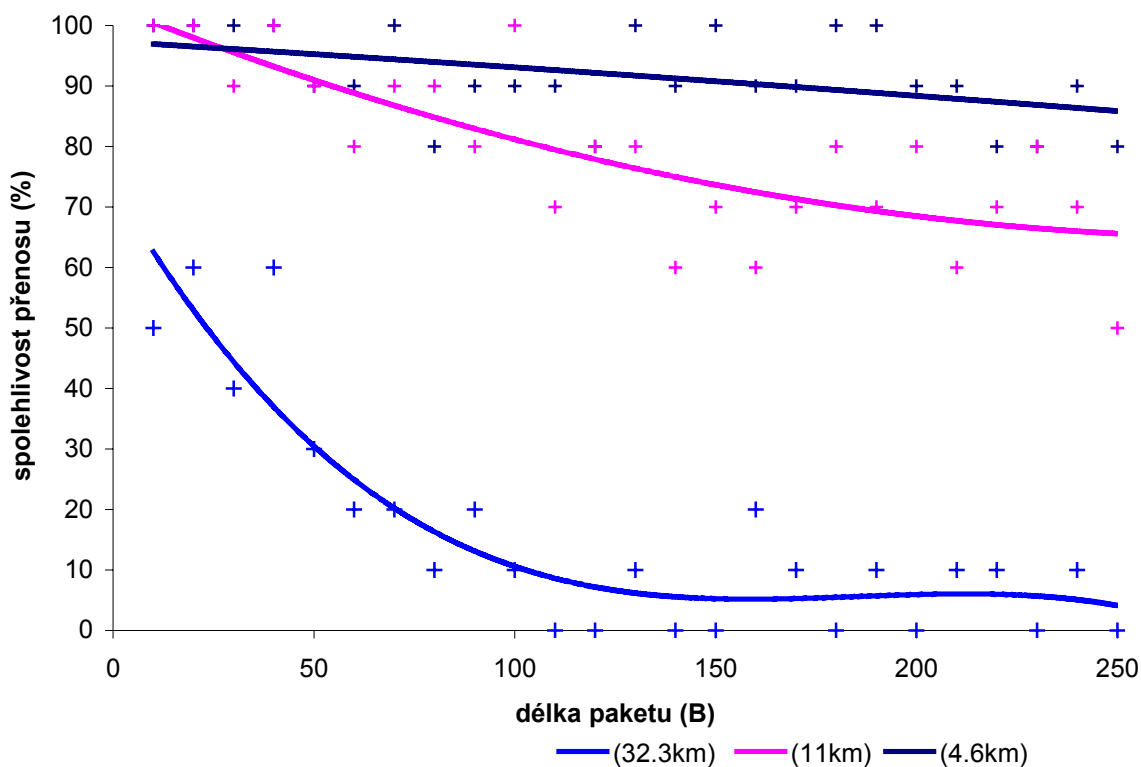


Obr. 20 Graf závislosti spolehlivosti přenosu na intenzitě okolního šumu

5.1.3. Vliv délky paketu na spolehlivost přenosu.

V kapitole 3.4. při tvorbě komunikačního protokolu byl zmíněn předpoklad vyšší spolehlivosti přenosu u kratších datových rámců. Cílem této kapitoly je pokročilejší analýza dané problematiky vedoucí k jejímu lepšímu objasnění.

Testy byly prováděny na stejných trasách jako u předchozího testu za podmínek okolního rušení v rozmezí 7S až 9S. Přenášen byl řetězec o příslušné délce skládající se z řetězců „1234567890“ v délce 10 až 250 bytů po 10bytových krocích. Každý řetězec byl odeslán desetkrát. Vzhledem k použité metodě měření není výsledek 100% objektivní, neboť mezi úspěšně přenesené rámce nejsou zahrnuty rámce ztracené na zpáteční cestě. Výsledný trend spolehlivosti je uveden na obrázku 21. Naměřené hodnoty byly lehce rozkolísané, proto jsou proloženy regresní polynomiální křivkou, jež lépe vystihuje statistický charakter spolehlivosti přenosu.



Obr. 21 Graf závislosti spolehlivosti přenosu na délce přenosového rámce

Hlavní činitelé ovlivňující trend spolehlivosti přenosu jsou drobné kolize a interference na přenosovém pásmu vznikající při vysílání datových rámců vzdálených stanic. Tento problém úzce souvisí s charakterem sdílení přenosového média u všesměrové topologie sítě, o níž pojednávala kapitola 2.3.1. Vzdálená stanice nemusí přímo při své komunikaci s druhou vzdálenou stanicí vytvářet lokální kolize, avšak jejich komunikace může způsobit interferenci s odesílaným rámcem vzdálené stanice komunikující na velkou vzdálenost.

Z uvedeného grafu je patrné, že bezdrátová síť CB packet radia je dostatečně spolehlivá při přenosech na krátké vzdálenosti, čímž se hodí zejména do městského prostředí. Při potřebě překlenutí větší vzdálenosti je vhodné optimalizovat umístění vysílače s anténou a zamezit interferencím mezi lokální a vzdálenou komunikací volbou rozdílných přenosových kanálů.

5.2. Výhody a nevýhody sběru technologických dat sítí na bázi TNC

Testovaná síť paket radia na bázi TNC vykazuje lehce odlišné chování oproti v průmyslu běžně používaným komunikačním standardům. Je to dáno především rozdílnými záměry tvůrců prostředků pro komunikaci v síti s protokolem AX.25, jenž byly směřovány k civilnímu nasazení. Je nutné pečlivě zvážit oblasti a účel možného nasazení této technologie v průmyslu, v nichž by byly potlačeny nevýhody a zdůraznily se výhody.

Výhody bezdrátové komunikace na bázi TNC jsou především v nízkých nákladech na provoz technologie, které se skládají převážně z pořizovacích nákladů. Vlastní provozní náklady pak představují pouze výdaje za energii. Vzhledem k masovému civilnímu rozšíření radiostanic pro pásmo CB, taktéž PR modemů, jsou ceny jednotlivých částí vybavení stanice stlačeny cenově do oblasti spotřební elektroniky. Nespornou výhodou je také otevřenost dokumentace, konstrukcí a jednotlivých zapojení, což umožňuje integraci modemu typu TNC přímo do PLC a nevyužívat komerční produkty, čímž se pořizovací náklady dále sníží. Odhadované náklady na vybavení jedné stanice jsou v následující tabulce.

Položka	Cena (Kč)
CB stanice	1900
TNC	3000
Anténa	500
Zdroj	500
celkem	5900

Tabulka 2 Odhadovaná výše nákladů na realizaci jedné stanice

Nevýhody tohoto řešení lze spatřit především v nízké datové propustnosti sítě, nedostatečné možnosti utajení obsahu přenášené informace z důvodu otevřenosti formy přenosu. S tím souvisí také možnost snadného ovlivňování ostatními provozovateli sítě PR na stejném kanále. V případě koncepce přenosu dat použité v diplomové práci vzniká rovněž otázka, do jaké míry jsou přijatelné ztráty paketů nesoucí informaci. Za použití spojovaného režimu odpadá problém se spolehlivostí přenosu, avšak je řídicími rámci nutnými pro udržení spojení zvýšené zatížení přenosového kanálu, což v konečném důsledku omezí maximální množství stanic v síti, nebo četnost přenášení dat.

Problematickým prvkem celé koncepce se může stát implementační náročnost programu obsluhy TNC do PLC protokolem HOSTMODE. V jistém ohledu lze uvažovat o zjednodušení využitím levnějších TNC postavených nad jednodušším komunikačním protokolem KISS. Dále pro úlohu prostého zasílání telemetrických dat bez potřeby zpětné vazby lze problém dále zjednodušit vyřazením TNC z celé koncepce a nahrazením jednodušší a levnější formou modemu, či přímo samostatným modulátorem. V posledním uvedeném zjednodušení by se přenášený paket skládal přímo v PLC a odesílal modulátoru umístěném na portu RS232. Každé zjednodušení však snižuje oblast využití a užitnou hodnotu celého řešení.

Uvedenou technologii lze s úspěchem nasadit v situaci, kdy je kladen důraz na levný systém přenosu dat z měřicích míst v okruhu do 10 km od centrální stanice za podmínek, že nejsou kladeny příliš vysoké nároky na spolehlivost a celkový objem přenesených dat. Ztrátovost části dat lze kompenzovat častějším odesíláním informace. Systém je vhodné používat pro účely monitorování rozsáhlých pomalých technologií.

6. Závěr

Přenos a zpracování technologických dat je klíčovým prvkem v moderním pojetí řízení průmyslových technologií. V této oblasti existuje bohatá nabídka produktů určených pro řízení, přenos a zpracování dat u technologií, které jsou prostorově soustředěny do malé oblasti. Firmy užívající technologie rozptýlené do rozsáhlé oblasti nemají tak velký výběr a jsou často nuceny vyvíjet vlastní koncepci přenosu dat mezi jednotlivými částmi technologie.

V předkládané diplomové práci se nachází analýza problematiky a její komplexní řešení v podobě realizovaného monitorovacího systému, které obsahuje omezenou možnost řízení a je určeno pro pomalou rozptýlenou technologii. Pro datové přenosy je použita nízkorychlostní bezdrátová komunikace odvozená od koncepce civilní sítě packet radia a postavená na bázi modemu typu TNC.

Ačkoliv je síť packet radia koncipovaná jako spojovaná služba, bylo při vývoji softwarových prostředků pro ověření funkce bezdrátové komunikace experimentálně použito nespojovaného přenosu dat. Na základě zjištěného chování lze považovat tento nestandardní mechanismus používání v některých případech za vhodnější. Detailní popis chování sítě v tomto režimu je obsažen v kapitole 5.

Diplomová práce přináší dosud nepublikovaný příklad realizace a unikátního využití technologie sítě packet radia pro přenos technologických dat, čímž nabízí pro specifický okruh zájemců levnou a funkční alternativu k jejich soukromému či drahému zakázkovému vývoji. Softwarové prostředky byly vyvíjeny primárně pro monitorování technologie výměňkových stanic, ale vzhledem k celkové koncepci programu je možná snadná adaptace na jinou technologii.

Ideálním završením diplomové práce by bylo aplikování jejích dosavadních výsledků do praxe. I přes kompaktnost koncepce uvedené v této práci by bylo vhodné na ni navázat a téma dále rozvíjet. Jednou z perspektivních cest představuje samostatný vývoj TNC založený na soudobé součástkové základně s následnou možností experimentování na poli zvyšování přenosové rychlosti, nebo robustnosti přenosu dat, volbou některé z moderních forem modulace signálu.

7. Seznam použité literatury

- [1] Václavík Radek, Lajsner Pavel: *Packet radio od A skoro až do Z*, BEN, Praha 1996
- [2] Doc. Ing. Plášil František, CSc., Doc. Ing. Staudek Jan, CSc. a kol.: *Ročenka výpočetní techniky 1 – Informatika*, SNTL Praha 1989
- [3] *Firemní manuál k TNC-2 STABO*
- [4] *Paket rádio, nejčastější provoz radioamatérů*, Amatérské Rádio – Praktická elektronika 11/2001 až 5/2002
- [5] *Občanské radiostanice CB v praxi*, Amatérské Rádio – Řada B, 3/1994
- [6] Jaroslav Janda: *Jemný úvod do SQL*, Czech-Computer 1997
- [7] Borland IDD: *Delphi Database Application Developer's Guide*
- [8] Borland Software Corporation: *Delphi 6 for Windows – Developer's Guide*, Borland, 2001
- [9] DuBois Paul: *MySQL profesionálně: komplexní průvodce použitím, programováním a správou MySQL*, Mobil Media, Praha 2003

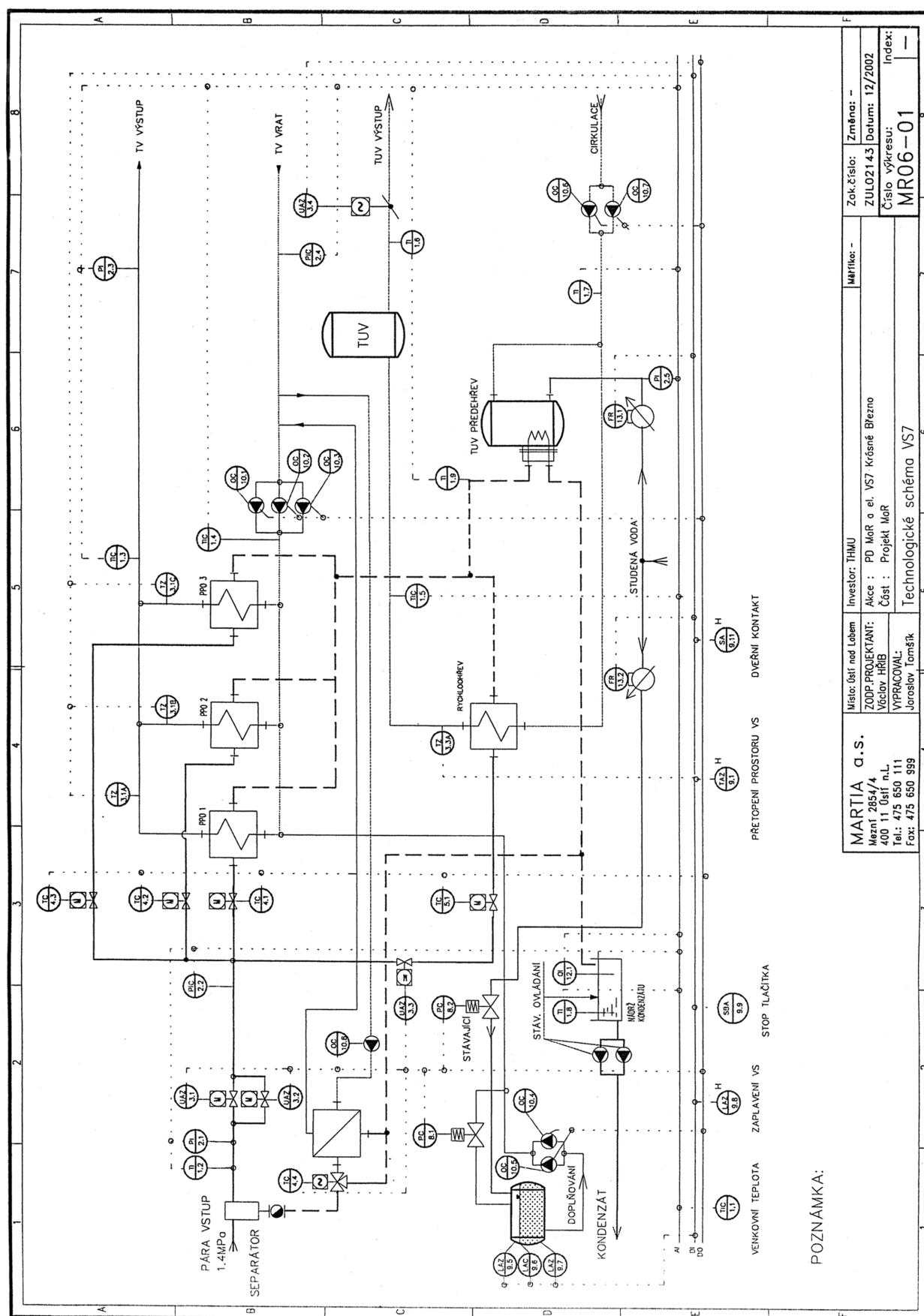
Internetové stránky

- [10] *Historie Packet-radia*: <http://www.tapr.org/tapr/html/Fpetfaq.html>, říjen 2003
- [11] *Popis protokolu AX.25*: <http://www.qsl.net/ok0ns/AX25/AX25.TXT>, říjen 2003
- [12] *Historie ARPANETu*: <http://www.dei.isep.ipp.pt/docs-arpa.html>, říjen 2003
- [13] *Popis protokolu X.25*: <http://www2.rad.com/networks/1996/x25/x25.htm>, leden 2004
- [14] *Specifikace HOSTMODE dle WA8DED*: <http://www.nordlink.org/eng/wa8host.txt>, říjen 2002
- [15] *Popis THE-FIRMWARE verze 2.7*: <http://www.nordlink.org/tf27doc.lhz>, říjen 2002
- [16] *Příkazy terminal-mode*: http://www.raflzimmermann.de/tf26_2.html, říjen 2002
- [17] *Zdrojový kód komponenty QCCom32*:
<http://geocities.com/scottpinkham/delphi/delphi.html>, říjen 2003

8. Seznam příloh

PŘÍLOHA A – BLOKOVÉ SCHÉMA VÝMĚNÍKOVÉ STANICE.....	61
PŘÍLOHA B – TABULKA OSAZENÍ REGULÁTORU SENZORY	62
PŘÍLOHA C – DATOVÝ MODEL APLIKACE	66
PŘÍLOHA D – VZHLED VYTVOŘENÉ APLIKACE	68
PŘÍLOHA CD se zdrojovými kódy, binárními kódy aplikace a příslušenstvím	obálka na zadních deskách

Příloha A – Blokové schéma výměníkové stanice



Příloha B – Tabulka osazení regulátoru senzory

Níže uvedené tabulky pocházejí z projektové dokumentace regulace výměňkové stanice VS7.

Základní deska IRMS 2000 15AI, 8AO, 8BI, 8BO

TYP	POZICE	ADRESA	SV	OKRUH	FUNKCE
PŘIPOJENÍ	18V AC	X1:	1		Napájení, trafo 18V AC
	18V AC		2		Napájení, trafo 18V AC
	24V DC -		3		Záložní modul napájení, -24V DC
	24V DC -		4		
	24V DC -		5		
	24V DC +		6		Záložní modul napájení, +24V DC
	24V DC +		7		
	24V DC +		8		
	INT -		9		Záložní modul, -24V DC
	INT+		10		Záložní modul, signál - nouzové napájení
ANALOGOVÉ VÝSTUPY	24V DC -		11		Společný - externí napájení
	24V DC +		12		Společný - externí napájení
	OUT0	0	13	11.I	Regulace FM (380V, 8.3A, 0-10V)
	OUT1	2	14		
	OUT2	4	15		
	OUT3	6	16		
	OUT4	8	17		
	OUT5	10	18		
	OUT6	12	19		
	OUT7	14	20		
ANALOGOVÉ VSTUPY	24V DC -		21		Společný - externí napájení
	24V DC +		22		Společný - externí napájení
	INP0	4860	23	T1.1	Venkovní teplota
	INP1	4862	24	T1.3	Teplota TV za výměníky
	INP2	4864	25	T1.4	Teplota TV vratu
	INP3	4866	26	T1.2	Teplota páry před redukcí
	INP4	4868	27	T1.8	Teplota kondenzátu
	INP5	4870	28	Q12.1	Měření vodivosti kondenzátu
	INP6	4872	29	T1.5	Teplota TUV za výměníkem
	NC		30		
	24V DC -		31		Společný - externí napájení
	24V DC +		32		Společný - externí napájení
	INP0	4874	33	T1.6	Teplota TUV na výstupu
	INP1	4876	34	P2.1	Tlak páry na vstupu
	INP2	4878	35	P2.2	Tlak páry za redukcí
	INP3	4880	36	T1.9	Teplota kondenzátu před přehřevem
	INP4	4882	37	T1.7	Teplota TUV cirkulační
	INP5	4884	38	P2.5	Tlak studené vody
	INP6	4886	39	P2.3	Tlak TV za výměníky
	INP7	4888	40	P2.4	Tlak TV vratu

TYP	POZICE	ADRESA	SV	OKRUH	FUNKCE
BINÁRNÍ VÝSTUPY	24V DC -		41		Společný - externí napájení
	24V DC +		42		Společný - externí napájení
	OUT0	16.0	43	UAZ3.1-2	HUV, redukce tlaku na vstupu do VS - OTEVÍRÁ
	OUT1	16.1	44	UAZ3.1-2	HUV, redukce tlaku na vstupu do VS - ZAVÍRÁ
	OUT2	16.2	45	HL9.12	Porucha TUV
	OUT3	16.3	46	UAZ3.1-2	HUV, redukce tlaku na vst. Do VS - HAV.ZAVÍRÁ
	OUT4	16.4	47	HL9.11	Porucha VS
	OUT5	16.5	48	CO10.8	Čerpadlo separátoru
	OUT6	16.6	49	TC4.4	Servopohon separátoru
	OUT7	16.7	50	PC8.3	Sol. Ventl odpouštění TS
BINÁRNÍ VSTUPY	INP0	4956.0	51	UAZ3.4	Klapka TUV (Sig. Otevřena/Zavřena)
	INP1	4956.1	52		Přetopení TV - havárie VS
	INP2	4956.2	53	L9.8	Zaplavení VS
	INP3	4956.3	54		Rezerva
	INP4	4956.4	55		Rezerva
	INP5	4956.5	56		Rezerva
	INP6	4956.6	57	L9.5	Hladina v doplňovací nádrži (Sig. Maxima)
	INP7	4956.7	58	L9.6	Hladina v doplňovací nádrži (Regulace)
	24V DC -		59		Společný - externí napájení
	24V DC +		60		Společný - externí napájení

Komunikační linky (verze: 5xCOM – 0115.11, 1xCOM – 0115.01)

TYP	POZICE	ADRESA	SV	OKRUH	FUNKCE
COM 0	+CP	(standardní komunikační linka)	61		Napájení komunikace +24V DC
	CP				rozhraní RS232 (Parametrování + displej)
	-CP		62		Napájení komunikace -24V DC
COM 1	+CA		61		Napájení komunikace +24V DC
	CA				
	-CA		62		Napájení komunikace -24V DC
COM 2	+CB		61		Napájení komunikace +24V DC
	CB				
	-CB		62		Napájení komunikace -24V DC
COM 3	+CC		61		Napájení komunikace +24V DC
	CC				
	-CC		62		Napájení komunikace -24V DC
COM 4	+CD		61		Napájení komunikace +24V DC
	CD				
	-CD		62		Napájení komunikace -24V DC

I/O deska 24BO, 24BI

TYP	POZICE	ADRESA	SV	OKRUH	FUNKCE
BINÁRNÍ VÝSTUPY	24V DC -		1		Společný - externí napájení
	24V DC +		2		Společný - externí napájení
	OUT0	17.0	3	PC8.4	Sol. Ventil dopouštění vody do doplň nádrže
	OUT1	17.1	4	OC10.1	Oběhové čerpadlo TV
	OUT2	17.2	5	OC10.2	Oběhové čerpadlo TV
	OUT3	17.3	6	OC10.3	Oběhové čerpadlo TV
	OUT4	17.4	7	OC10.6	Doplňovací čerpadlo TV
	OUT5	17.5	8	OC10.7	Doplňovací čerpadlo TV
	OUT6	17.6	9	11.I	Start FM
	OUT7	17.7	10		Rezerva
	24V DC -		11		Společný - externí napájení
	24V DC +		12		Společný - externí napájení
	OUT0	18.0	13		Rezerva
	OUT1	18.1	14		Rezerva
	OUT2	18.2	15	TC5.1-2	Regulace teploty TUV - OTEVÍRÁ
	OUT3	18.3	16	TC5.1-2	Regulace teploty TUV - ZAVÍRÁ
	OUT4	18.4	17	UAZ3.4	Uzavírací klapka na výstupu TUV z VS - O/Z
	OUT5	18.5	18	UAZ3.3	Uzav. Ventil na vst. do výměníku TUV - O/Z
	OUT6	18.6	19		Rezerva
	OUT7	18.7	20		Rezerva
	24V DC -		21		Společný - externí napájení
	24V DC +		22		Společný - externí napájení
	OUT0	19.0	23	OC10.4	Cirkulační čerpadlo TUV
	OUT1	19.1	24	OC10.5	Cirkulační čerpadlo TUV
	OUT2	19.2	25	TC4.1	Regulace teploty TV1 - OTEVÍRÁ
	OUT3	19.3	26	TC4.1	Regulace teploty TV1 - ZAVÍRÁ
	OUT4	19.4	27	TC4.2	Regulace teploty TV2 - OTEVÍRÁ
	OUT5	19.5	28	TC4.2	Regulace teploty TV2 - ZAVÍRÁ
	OUT6	19.6	29	TC4.3	Regulace teploty TV3 - OTEVÍRÁ
	OUT7	19.7	30	TC4.3	Regulace teploty TV3 - ZAVÍRÁ

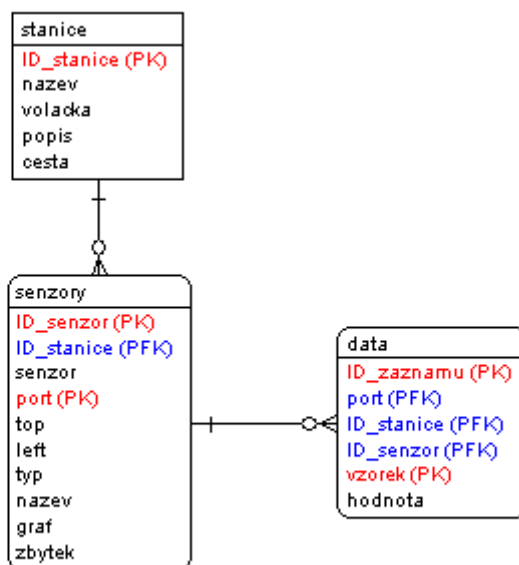
TYP	POZICE	ADRESA	SV	OKRUH	FUNKCE
BINÁRNÍ VSTUPY	24V DC -		31		Společný - externí napájení
	24V DC +		32		Společný - externí napájení
	INP0	4957.0	33	L9.7	Hladina v doplňovací nádrži (Sig. Minima)
	INP1	4957.1	34	T9.1	Přetopení prostoru VS
	INP2	4957.2	35	SB9.9	Stop Tlačítko
	INP3	4957.3	36	SA9.10	Kvitace poruchy
	INP4	4957.4	37	OC10.1	Oběhové čerpadlo TV - Sig CHOD/STOP
	INP5	4957.5	38	OC10.2	Oběhové čerpadlo TV - Sig CHOD/STOP
	INP6	4957.6	39	HAF	Výpadek fáze
	INP7	4957.7	40	OC10.3	Oběhové čerpadlo TV - Sig CHOD/STOP
	24V DC -		41		Společný - externí napájení
	24V DC +		42		Společný - externí napájení
	INP0	4958.0	43		
	INP1	4958.1	44	OC10.6	Doplňovací čerpadlo TV - Sig CHOD/STOP
	INP2	4958.2	45	OC10.7	Doplňovací čerpadlo TV - Sig CHOD/STOP
	INP3	4958.3	46		
	INP4	4958.4	47		FM v chodu (Rezerva)
	INP5	4958.5	48		Porucha FM (Rezerva)
	INP6	4958.6	49	UAZ3.3	Uzav. Ventil na vst. Do výměníku TUB - ZAVŘENO
	INP7	4958.7	50	OC10.4	Cirkulační čerpadlo TUV - Sig. CHOD/STOP
	INP0	4959.0	51	OC10.5	Cirkulační čerpadlo TUV - Sig. CHOD/STOP
	INP1	4959.1	52	FQ13.1	Vodoměr SV TUV
	INP2	4959.2	53	SA9.11	Kontakt dveří
	INP3	4959.3	54	FO13.2	Vodoměr doplňování TS
	INP4	4959.4	55		
	INP5	4959.5	56		
	INP6	4959.6	57		
	INP7	4959.7	58		
	24V DC -		59		Společný - externí napájení
	24V DC +		60		Společný - externí napájení

Příloha C – Datový model aplikace

Datový model aplikace byl vytvořen v programu CaseStudio verze 2.10.1 společnosti RK-Soft. Kompletní dokumentace k datovému modelu ve formátu HTML a RTF je uložena na příloženém CD v adresáři „Datový model“.

Entitně relační diagram

[1,1]



Seznam entit

Název entity	Typ entity	Primární klíč	Počet atr.
Data	závislá	ID_zaznamu, port, ID_stanice, ID_senzor, vzorek	6
Senzory	závislá	ID_senzor, ID_stanice, port	10
Stanice	nezávislá	ID_stanice	5

Entita 'stanice'

Název atributu/role	Datový typ	Not null	Unique	Popis
ID_stanice	Integer	ANO	ANO	Umělý klíč (autoinkrementální)
nazev	Char	NE	NE	Název stanice
volacka	Char	ANO	ANO	Volací znak stanice
popis	Text	NE	NE	Doplňující informace ke stanici
cesta	Char	NE	NE	Směrovací řetězec

Entita 'data'

Název atributu/role	Datový typ	Not null	Unique	Popis
ID_zaznamu	Bigint	ANO	ANO	Umělý klíč (autoinkrementální)
port	Tinyint	ANO	NE	číslo portu
ID_stanice	Integer	ANO	NE	
ID_senzor	Integer	ANO	NE	hodnota se nepoužívá
vzorek	Integer	ANO	NE	Hodnota datumu a času v 5min vzorkování
hodnota	Integer	NE	NE	Numerická hodnota informace. Pokud senzor vyžaduje hodnoty True/False, jsou použity 1/0

Indexy

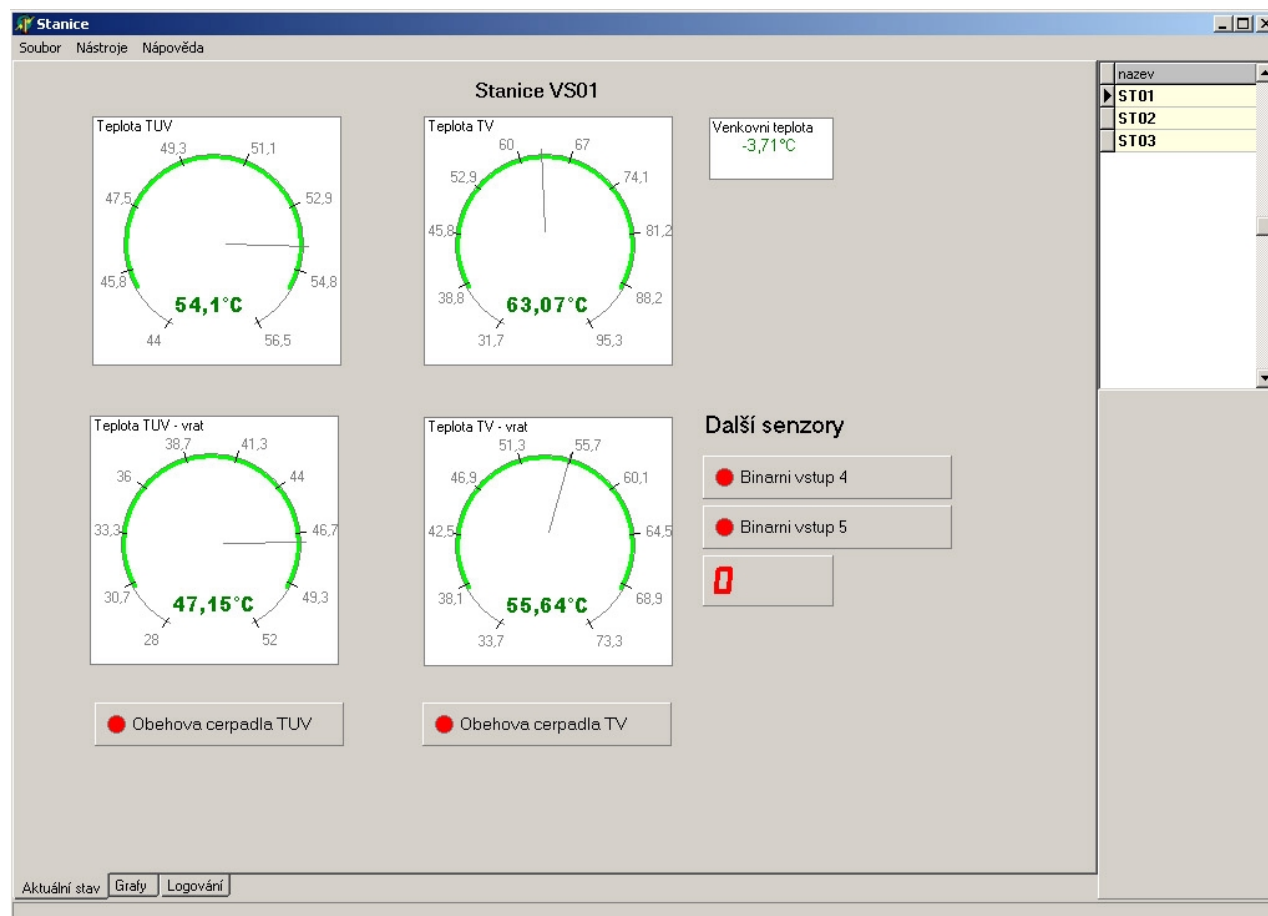
Název indexu	Atributy	Unique	Vzestupný	Clustrovaný
třídění	vzorek, port, ID_stanice	NE	NE	NE

Entita 'senzory'

Název atributu/role	Datový typ	Not null	Unique	Popis
ID_senzor	Integer	ANO	ANO	Umělý klíč (autoinkrementální)
ID_stancie	Integer	ANO	NE	Umělý klíč.
senzor	Char	NE	NE	Typ senzoru - ANALOG / DIGITAL / COUNTER / LABEL
port	Tinyint	ANO	NE	Přístupový port senzoru.
top	Integer	NE	NE	Poloha senzoru na ploše.
left	Integer	NE	NE	Poloha senzoru na ploše.
typ	Tinyint	NE	NE	Vizuální forma vybraného senzoru.
nazev	Char	NE	NE	Orientační název senzoru.
graf	Bool	NE	NE	Má se vykreslovat graf ?
zbytek	Char	NE	NE	Ostatní parametry nastavení senzoru.

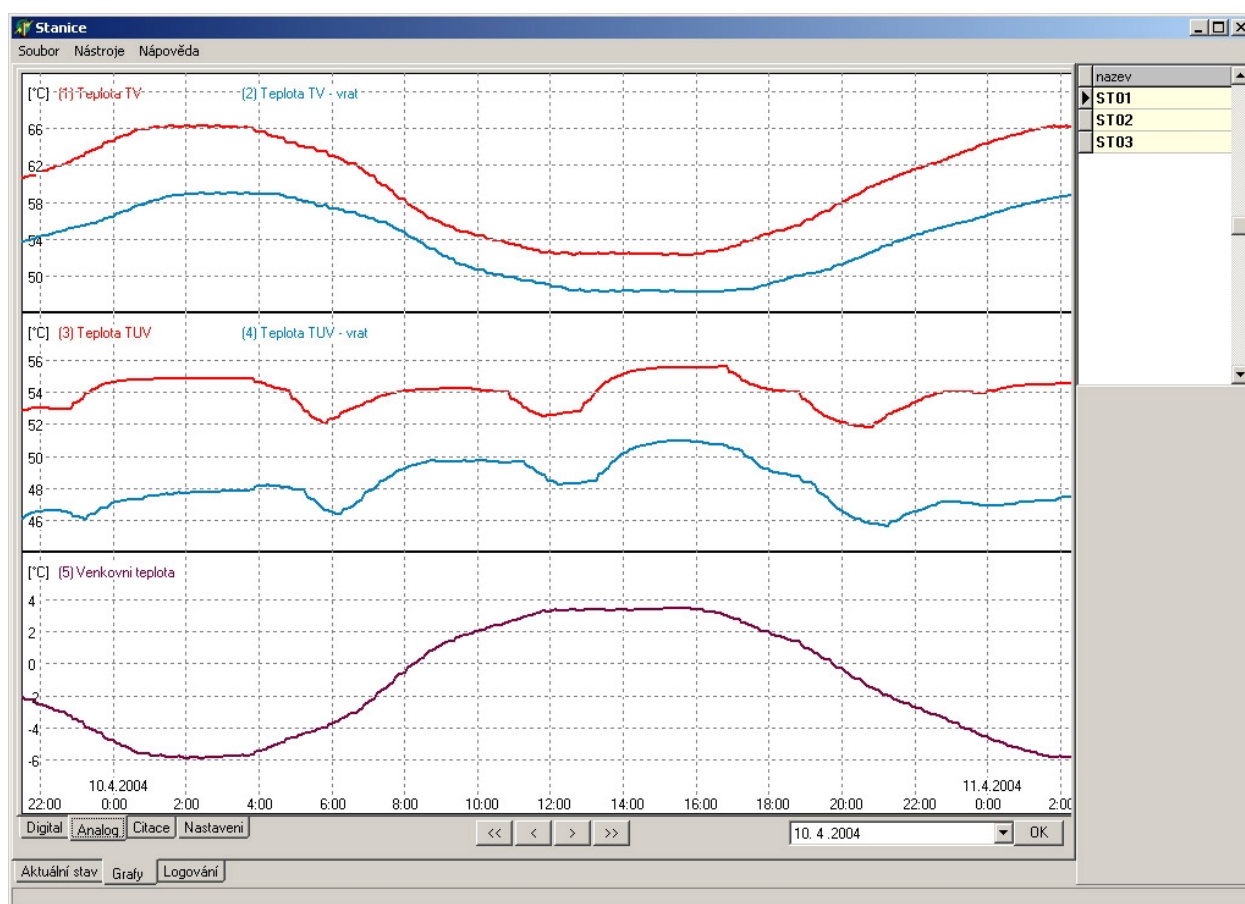
Příloha D – Vzhled vytvořené aplikace

Na obrázku č. 22 je zachycena základní kompozice vytvořené aplikace. V levé části se nachází volitelné listy zobrazovacího režimu. V uvedeném případě je aktivní list „Aktuální stav“, na němž jsou rozmístěny „inteligentní komponenty“. V pravé části je výběr sledované stanice. Další možnosti velké zobrazovací plochy jsou na listech „Grafy“ a „Logování“.



Obr. 22 Plocha „Aktuální stav“

Na obrázku č. 23 je aktivní list „Grafy“. V tomto listu jsou umístěny listy pro grafickou reprezentaci jednotlivých typů sledovaných hodnot. Na obrázku č. 23 je zobrazena grafická interpretace hodnot z analogových portů. Pole tlačítek „<< < > >>“ pod grafem slouží k pohybu po časové ose o jeden den (platí pro tlačítka „< >“), nebo o celý zobrazený časový úsek (platí pro tlačítka „<< >>“). Pro posun na konkrétní den slouží datumové pole s kalendářem.



Obr. 23 Průběh stavů získaných z analogových senzorů

Průběh zaznamenaných analogových stavů zobrazený na obrázku č. 23 byl zvolen tak, aby bylo možné poukázat na význam parametru „skupina“ analogového inteligentního senzoru. Hodnoty, které mají jistou souvislost, jsou zobrazeny do společného grafu. Jednotlivé osy grafů jsou algoritmem zvoleny tak, aby nebyly ani řídké ani husté, pokrývaly vymezený rozsah hodnot a zároveň spadaly do oboru celých čísel.

Na obrázku č. 24 je vybrán výřez s obsahem listu „nastavení“. Tento list ovlivňuje nastavení vykreslování na ostatních listech z nabídky grafů. Pokud pro vybranou stanici existuje sledovaný senzor, bude zde aktivní. Vybrané aktivní senzory jsou následně vykresleny v grafech.

Zobrazovat analogové hodnoty z portů

<input checked="" type="checkbox"/> port 1	<input checked="" type="checkbox"/> port 5	<input type="checkbox"/> port 9	<input type="checkbox"/> port 13
<input checked="" type="checkbox"/> port 2	<input type="checkbox"/> port 6	<input type="checkbox"/> port 10	<input type="checkbox"/> port 14
<input checked="" type="checkbox"/> port 3	<input type="checkbox"/> port 7	<input type="checkbox"/> port 11	<input type="checkbox"/> port 15
<input checked="" type="checkbox"/> port 4	<input type="checkbox"/> port 8	<input type="checkbox"/> port 12	

Zobrazovat hodnoty stavu čítačů z portů

<input checked="" type="checkbox"/> port 1	<input type="checkbox"/> port 5	<input type="checkbox"/> port 9	<input type="checkbox"/> port 13
<input type="checkbox"/> port 2	<input type="checkbox"/> port 6	<input type="checkbox"/> port 10	<input type="checkbox"/> port 14
<input type="checkbox"/> port 3	<input type="checkbox"/> port 7	<input type="checkbox"/> port 11	<input type="checkbox"/> port 15
<input type="checkbox"/> port 4	<input type="checkbox"/> port 8	<input type="checkbox"/> port 12	

Zobrazovat stavy digitalních portů

<input checked="" type="checkbox"/> port 1	<input type="checkbox"/> port 9	<input type="checkbox"/> port 17	<input type="checkbox"/> port 25
<input checked="" type="checkbox"/> port 2	<input type="checkbox"/> port 10	<input type="checkbox"/> port 18	<input type="checkbox"/> port 26
<input type="checkbox"/> port 3	<input type="checkbox"/> port 11	<input type="checkbox"/> port 19	<input type="checkbox"/> port 27
<input checked="" type="checkbox"/> port 4	<input type="checkbox"/> port 12	<input type="checkbox"/> port 20	<input type="checkbox"/> port 28
<input checked="" type="checkbox"/> port 5	<input type="checkbox"/> port 13	<input type="checkbox"/> port 21	<input type="checkbox"/> port 29
<input type="checkbox"/> port 6	<input type="checkbox"/> port 14	<input type="checkbox"/> port 22	<input type="checkbox"/> port 30
<input type="checkbox"/> port 7	<input type="checkbox"/> port 15	<input type="checkbox"/> port 23	<input type="checkbox"/> port 31
<input type="checkbox"/> port 8	<input type="checkbox"/> port 16	<input type="checkbox"/> port 24	<input type="checkbox"/> port 32

Preferované zobrazení grafů

☒ Tento den
☐ Posledních 24 hodin
☐ Posledních 48 hodin
☐ Posledních 7 dní

Digital Analog Citace **Nastavení** << < > >> 10. 4. 2004 OK

Obr. 24 Nastavení parametrů vykreslování

Na obrázku č. 25 je zobrazena základní podoba programu pro obsluhu TNC, konkrétně části serverové. Velké textové pole v centrální oblasti obsahuje data získaná z virtuálního monitorovacího kanálu TNC. Tlačítka na horní liště slouží k manuálnímu vytvoření spojení a konfiguraci programu. ProgressBary po obou stranách umožňují vizuální kontrolu stavů naplnění vstupních (vlevo) a výstupních (vpravo) front a bufferů.



Obr. 25 Grafická podoba programu obsluhy TNC